



## **PORTAL-FORO PARA EL APRENDIZAJE DE LENGUAS HISPANO-JAPONESAS**

Memoria del Proyecto Fin de Carrera  
de Ingeniería en Informática  
realizado por  
Michael-Jorge Gómez Campos  
y dirigido por  
Pilar Gómez Sánchez  
Bellaterra, 18 de Septiembre de 2009

*Dedico esta memoria a Makiko  
por ofrecerme la oportunidad de participar en este proyecto*

*También a Pilar  
por acoger el proyecto y ayudarme a lo largo del camino*

*Y también se lo dedico a mis padres y a Dailín,  
por apoyarme, animarme y forzarme a ir al día*

*Muchas gracias a todos*

# ÍNDICE DE CONTENIDOS

|  |    |
|--|----|
| 1. Introducción al proyecto.....                                     | 6  |
| 1.1 Motivación del proyecto.....                                     | 6  |
| 1.2 Objetivos del proyecto.....                                      | 10 |
| 1.3 Organización de la memoria.....                                  | 10 |
| 2. Estado del arte y planificación de tareas.....                    | 12 |
| 2.1 Estado del arte.....   | 12 |
| 2.1.1 E-Learning.....  | 12 |
| 2.2 Análisis de recursos.....  | 14 |
| 2.2.1 LMS (Learning Managemenr System).....                          | 14 |
| 2.2.1.1 Sistemas LCMS de pago.....                                   | 15 |
| 2.2.1.2 Sistemas LCMS Open-Source.....                               | 16 |
| 2.2.1.3 Conclusión.....  | 17 |
| 2.3 Planificación y organización de tareas.....                      | 18 |
| 3. Plataforma y herramientas de desarrollo.....                      | 20 |
| 3.1 Plataforma Moodle.....   | 20 |
| 3.1.1 Requisitos de servidor.....                                    | 20 |
| 3.1.1.1 Disco duro.....  | 20 |
| 3.1.1.2 Memoria RAM.....   | 21 |
| 3.1.1.3 CPU.....   | 21 |
| 3.1.1.4 Red de conexión.....   | 22 |
| 3.1.2 Requisitos de los usuarios.....                                | 22 |
| 3.2 Herramientas de desarrollo.....                                  | 22 |
| 3.2.1 XAMPP.....   | 22 |
| 3.2.2 Aptana Studio.....   | 22 |
| 3.2.4 jQuery.....  | 23 |
| 3.2.4 Adobe Flash.....   | 23 |
| 4. Diseño e implementación.....                                      | 25 |
| 4.1 Primera fase: Preparación de la plataforma Moodle.....           | 25 |
| 4.2 Segunda fase: Creación del módulo de aprendizaje de japonés..... | 27 |
| 4.2.1 Arquitectura de los módulos.....                               | 28 |
| 4.2.2 block_nihongo.....   | 30 |
| 4.2.2.1 Palabra del Momento.....                                     | 32 |
| 4.2.3 Bloque principal (mod_nihongo).....                            | 33 |
| 4.2.3.1 Tarjetas de estudio de kanji (cardvw).....                   | 34 |
| 4.2.3.2 Listado de kanji (kanjivw).....                              | 37 |
| Configurador del listado de kanji (block_kanji_config).....          | 39 |
| Administrador de listas de usuario (block_userlist_nihongo).....     | 40 |
| 4.2.3.3 Pizarra de kanji (trainvw).....                              | 43 |
| La pizarra Flash.....  | 45 |
| La pantalla de la pizarra.....                                       | 47 |
| Arquitectura del script de servidor.....                             | 48 |
| 5. Resultados y conclusiones.....                                    | 50 |
| 5.1 Pruebas realizadas y resultados.....                             | 50 |
| 5.2 Conclusiones y mejoras.....                                      | 51 |
| 6. Anexos.....   | 53 |
| ANEXO 1: Escritura japonesa.....                                     | 53 |
| ANEXO 2: Diccionario de datos.....                                   | 54 |

|   |    |
|---|----|
| KANJIDIC2.....                                  | 55 |
| JMdict.....                                     | 55 |
| ANEXO 3: Modificaciones internas de Moodle..... | 56 |
| 7. Bibliografía y referencias.....              | 61 |

# 1. Introducción al proyecto

En este capítulo se expondrán los detalles de la motivación, los objetivos y la organización de la memoria del proyecto.

## 1.1 Motivación del proyecto

En el año 2006, varios profesores de tres universidades, tanto de enseñanza de lengua japonesa (profesora Makiko Fukuda, de la Universidad Autónoma de Bellaterra (UAB) y profesora Yuki Suzuki, de la universidad Complutense de Madrid (UCM)) como de lengua española (Hiroko Terada, de la Keio University Shonan Fujisawa Campos (Keio), de Fujisawa, Japón) decidieron reforzar la enseñanza de sus clases con la creación de un foro conversacional en el que profesores y alumnos propusieran temas de conversación sobre temas familiares, rutina diaria o cultura general, con tal de *reforzar las habilidades de la comunicación práctica bajo el CEFR (Common European Framework of Reference for Languages: Learning, Teaching, Assessment)*, [39] y conseguir que los alumnos pudiesen llegar a alcanzar el nivel A2 del conocimiento de la lengua, nivel de conocimiento elemental del idioma según el CEFR.

El primer foro se creó para el curso 2006-07, que en Japón correspondería al último semestre del curso 2006-07 y el primer semestre del curso 2007-08. Fue programado en *Perl*, lenguaje de programación *shell*, que genera *scripts* en CGI (*Common Gateway Interface* - Interfaz de entrada común), tecnología que permite al cliente solicitar datos de un programa ejecutado en un servidor web. La combinación *Perl* + CGI es lo más utilizado en aplicaciones web japonesas, la mayoría sobre un servidor con *Unix* de soporte.

Desde la primera versión, se decidió la distribución de los alumnos en grupos pequeños, un foro por grupo. Se distribuyó a la gente en grupos de 15 personas, aproximadamente, llegando a reunir 13 grupos en total. Cada alumno accedía a su grupo con una cuenta de grupo única con la que no tenía acceso al resto de grupos. La participación fue bastante buena pero, a nivel de diseño, ofrecía una usabilidad muy pobre.

En esta primera versión, no existía la posibilidad de que el usuario tuviera una cuenta de usuario propia, propiciando la creación de temas de forma anónima. El formulario de creación de temas (ver figura 1) no dejaba claro cómo emplear los campos mostrados ni especificaba cuáles tenían que rellenarse de forma obligatoria ni ayuda alguna que facilitara su finalidad. El campo de contraseña era opcional y su función era registrar a un usuario anónimo como autor único de una respuesta o de un tema y se le ofrecía la posibilidad de poder modificar o eliminar su creación en un futuro.

**nombre**   
**email**   
**título**  **submit** **reset**  
**comentarios**  
  
**URL**   
**file**  No se h...archivo  
**contraseña**   
**color** ☒ abc ☐ abc ☐ abc ☐ abc ☐ abc ☐ abc ☐ abc ☐ abc

**Figura 1 - Formulario de creación de temas de la primera versión del foro**

Los temas de conversación (ver figura 2) se presentaban en diferentes bloques, uno seguido del otro, incluyendo las respuestas de los temas, éstas separadas por una línea divisora, mostrando cinco temas por página.

**Muerte de cantante de Zard** **Michael Jorge Gómez Campos** - 2007/05/29(Tue) 22:44 No.135 **responder**  
 こんにちは、皆さん。元気ですか？今日、ウェブページで「Zard」の歌手も死んだことを読みました！残念ですね。:(日本ではこの問題は大変でしたか？  
 Buenos días a todos. Cómo estáis? Hoy he leído en una web que murió la cantante de Zard! Menuda desgracia :( En japon ha tenido mucha importancia esta noticia?  
  
**Re: Muerte de cantante de Zard** **Natsuki Ohahiro** - 2007/05/03(Sun) 10:52 No.136  
 とっても大きな問題になっています。  
 私も悲しかったです。  
 Es la problema muy grande.  
 Estoy muy triste también.  
  
**こんばんは、みんな！** **Michael Gómez Campos** - 2007/05/10(Thu) 07:21 No.121 **responder**  
 こんにちは、皆さん！お久しぶりです！元気ですか？大学の授業の初めはよかったですが、今、たくさん試験がありますので、私は多忙です。明日、また試験があります。何かを聞。

**Figura 2 - Estructura de un tema de la primera versión del foro**

Cada tema disponía de un botón para *responder*, que se encargaba de abrir una nueva página mostrando el tema con sus respuestas y, debajo, un formulario como el de crear temas (ver figura 3).

El mayor problema venía dado por la falta de traducción de la mayoría de mensajes de navegación (ver en la figura 3 arriba) o por la mal traducción al castellano, dificultando la navegación a los estudiantes que estudiaban japonés. Un ejemplo de mala traducción se puede observar en la figura 1: el campo *comentarios* es una traducción literal de *comment*, en lugar de *Texto del mensaje* o similar, ocasionando malas interpretaciones de la funcionalidad del campo.

[戻る]

- 以下は、記事NO. 135に関する 返信フォーム です -

**Muerte de cantante de Zard Michael-Jorge Gómez Campos, 2007/05/29(Tue) 22:44 No.135**

こんにちは、皆さん。元気ですか？ 今日、ウェブページで「Zard」の歌手を死んだことを読みました！ 残念ですね。(日本ではこの問題は大切でしたか？)

Buenos dias a todos. Cómo estáis? Hoy he leído en una web que murió la cantante de Zard! Menuda desgracia :( En japon ha tenido mucha importancia esta noticia?

**Re: Muerte de cantante de Zard Natsuki Ohshiro, 2007/06/03(Sun) 10:52 No.136**

とっても大きな問題になっています。  
私も悲しかったです。

Es la problema muy grande.  
Estoy muy triste también.

nombre

email

título

comentarios

URL

file  No se h...archivo

contraseña

color ☐ abc ☐ abc ☐ abc ☐ abc ☐ abc ☐ abc ☐ abc ☐ abc

**Figura 3 - Estructura de la pantalla de respuesta de un tema de la primera versión del foro**

Para el curso siguiente se rediseñó el foro con la intención de hacerlo más usable y, por cuestiones internas, se ideó con la intención de ser traspasado a servidores de la UAB. Para ello se escogió, como lenguaje de desarrollo, PHP (*PHP Hypertext Pre-processor*), más utilizado en occidente y soportado por los servidores de la UAB. Se desarrolló entonces un foro muy sencillo utilizando, como base de datos, archivos de texto para los mensajes de cada usuario.

El formulario de creación de temas y respuestas se diseñó más accesible (ver figura 4), con una organización más clara y las etiquetas de los campos traducidos al castellano, pero se dejó sin traducir la información de cada campo (si eran opcionales, para qué servían algunos campos, etcétera).

A pesar de la mejora anterior, la navegación por el nuevo foro era lenta. Cada tema se representaba en forma de árbol, mostrando solamente el encabezado de los mensajes (ver figura 5), y para ver los mensajes de un tema concreto, obligaba a abrirlos, con procesamiento completo de página (ver figura 6), haciendo tediosa la tarea de leer un tema completo.

Una forma de minimizar el problema anterior se ofrecía en el menú del foro (ver en la figura 5 arriba, en rosa), al acceder a la pantalla de *Mostrar todos los mensajes*. Aquí se mostraban los mensajes de una manera muy similar al ofrecido en el foro de la versión anterior (ver figura 7), pero solamente mostraba 10 mensajes por página, sin control de temas.

**Grupo 03**  
【 Re2: ¡Hola! 】への返信

お名前/Nombre

メール/E-Mail  (任意)

Web/Página Web  (任意)

パスワード/Contraseña  (記事の変更/削除に必要)

タイトル/Título del mensaje

本文/Mensaje 

> Hola! Que' tal? Haga nuestro el mejor para el examen!  
 > こんにちは! 元気? 明後日のテストに向けて頑張りましょう!  
 >

☐ HTMLタグを有効にする(使用可能タグ一覧)
 ☐ URL直リンクを有効にする

アイコン/Icono  [一覧]

ファイル/Archivo  No seleccionar (画像、音声、動画のみ) (ファイル最大サイズ:1024K) [ご注意]

Figura 4 - Formulario de respuesta de tema de la segunda versión del foro

count: 2148 [管理画面]

**Grupo 03**

【 RSS 1.0 】 Group1 Group2 Group3 Group4 Q&A アンケート/sencuesta

新規投稿/Mensaje nuevo | 最終に更新/Actualizar foro | 全体表示/Mostrar todos los mensajes | 検索/Buscar | HOME/Página principal 🔒は 送付あり

- ★ [NEW] ¡Hola! あい、 08/06/25(Wed) 20:03
  - [NEW] Re1: ¡Hola! まゆ 08/07/12(Sat) 21:06
    - [NEW] Re2: ¡Hola! (ナーいち 08/07/12(Sat) 23:58
  - [NEW] Re1: ¡Hola! みづい、 08/07/11(Fri) 12:51
    - [NEW] Re2: ¡Hola! あい、 08/07/11(Fri) 16:21
      - [NEW] Re3: ¡Hola! みづい、 08/07/11(Fri) 22:07
- ★ [NEW] ¡Encantada! みづい、 08/06/15(Sun) 00:29
  - [NEW] Re1: ¡Encantada! あい、 08/07/11(Fri) 17:58
  - [NEW] Re1: ¡Encantada! まゆみ 08/07/03(Thu) 17:37
    - [NEW] Re2: ¡Encantada! みづい、 08/07/10(Thu) 20:48
      - [NEW] Re3: ¡Encantada! まゆみ 08/07/10(Thu) 22:35
        - [NEW] Re4: ¡Encantada! みづい、 08/07/11(Fri) 02:36
          - [NEW] Re5: ¡Encantada! まゆみ 08/07/11(Fri) 02:51
            - [NEW] Re6: ¡Encantada! みづい、 08/07/11(Fri) 12:44
              - [NEW] Re7: ¡Encantada! まゆみ 08/07/12(Sat) 17:45
                - [NEW] Re8: ¡Encantada! みづい、 08/07/12(Sat) 18:15
                  - [NEW] Re9: ¡Encantada! まゆみ 08/07/13(Sun) 02:47
        - [NEW] Re1: ¡Encantada! まゆ 08/06/24(Tue) 16:10
          - [NEW] Re2: ¡Encantada! みづい、 08/06/27(Fri) 02:56
            - [NEW] Re3: ¡Encantada! まゆ 08/06/27(Fri) 21:58

Figura 5 – Temas en forma de árbol de la segunda versión del foro

★ ¡Hola! あい、 08/06/25(Wed) 20:03

- Re1: ¡Hola! まゆ 08/07/12(Sat) 21:06
  - Re2: ¡Hola! (ナーいち 08/07/12(Sat) 23:58
- Re1: ¡Hola! みづい、 08/07/11(Fri) 12:51
  - Re2: ¡Hola! あい、 08/07/11(Fri) 16:21
    - Re3: ¡Hola! みづい、 08/07/11(Fri) 22:07

**Re2: ¡Hola!** [返信]

30a.080625200319461155.1.1 08/07/12(Sat) 23:58 by ナーいち [変更] [削除]

【 Re1: ¡Hola! 】への返信

Hola! Que' tal? Haga nuestro el mejor para el examen!  
 こんにちは! 元気? 明後日のテストに向けて頑張りましょう!

Figura 6 - Mensaje en la segunda versión del foro





**Figura 7 – Pantalla con todos los mensajes de un tema mostrados de la segunda versión del foro**

Además, el objetivo de confidencialidad por grupos del foro se vio afectado: los alumnos podían acceder a cualquier grupo y participar en ellos.

Para el presente curso se planteó buscar algo nuevo, más potente y accesible, que ofreciera mejores posibilidades, tanto a alumnos como a profesores.

## 1.2 Objetivos del proyecto

Tras lo expuesto en el apartado anterior, se planteó el desarrollo de un portal web de uso fácil, intuitivo y potente en el que alumnos y profesores pudiesen intercambiar conocimientos, practicar la lengua de estudio y corregir los errores cometidos. Además, se ofrecería la posibilidad de realizar actividades de aprendizaje, como estudiar vocabulario con un método de control inteligente de aprendizaje, o ejercicios de escritura a mano alzada de caracteres japoneses, con reconocimiento inteligente de caracteres.

Los recursos que contendría el portal serían:

- Un controlador de grupos de alumnos.
- Un foro conversacional especial para grupos de personas.
- Un *chat* de conversación *online*.
- Recursos para el aprendizaje de la lengua japonesa: reconocimiento de caracteres japoneses *online*, listado de kanji, tarjetas de kanji, listado propio de kanji.
- Acceso a páginas de interés.
- Posibilidad de ofrecer a los alumnos material multimedia.

## 1.3 Organización de la memoria

A continuación se detalla como está dividida la memoria, lo que permitirá describir las fases por las que pasa el desarrollo del proyecto. En el capítulo 2 se explica el estado del arte del proyecto, así

como un análisis de diferentes LCMS (*Learning Content Management System* – Sistema de Gestión de Contenidos de Aprendizaje) y una explicación sobre el sistema escogido. En el capítulo 3 se explican las características necesarias para la correcta instalación de *Moodle*, así como una descripción de las diferentes herramientas utilizadas para el desarrollo del proyecto. En el capítulo 4 se documenta el diseño y el desarrollo de los distintos componentes que constituyen el producto solicitado. En el capítulo 5 se da una muestra de los resultados obtenidos, mejoras futuras y conclusiones extraídas tras la realización del proyecto. El capítulo 6 expone una serie de anexos útiles que ayudan a la comprensión en detalle de diferentes temas utilizados para la realización del proyecto, así como una guía de cambios de código interno de *Moodle*. En el capítulo final se documenta la bibliografía y referencias web utilizadas.

## 2. Estado del arte y planificación de tareas

En este capítulo se describe el estado del arte del proyecto, un análisis de los recursos disponibles, el sistema escogido y la planificación y organización de las tareas realizadas.

### 2.1 Estado del arte

El aprendizaje de lenguas extranjeras mediante la sola asistencia a clase puede resultar incompleto, obligando al estudiante a practicar en horas extra-escolares para poder reforzar sus conocimientos y mejorar su fluidez con el idioma. Si nos concentramos en el aprendizaje de lenguas extranjeras con un nivel de dificultad alto, como son los idiomas asiáticos (el mandarín, el japonés o el coreano), el alumno debe adaptarse a una lengua de raíz totalmente distinta a la lengua materna. El japonés es un idioma que va más allá, pues ni siquiera tiene una rama lingüística común a otros idiomas. Si además tenemos en cuenta el complejo modelo de escritura oriental adaptada por los países del este asiático, comprobamos que el alumno debe enfrentarse al estudio de una lengua con vocabulario, gramática y escritura compleja de asimilar solamente en horario de clase.

#### 2.1.1 E-Learning

Una de las herramientas con las que puede contar un alumno en horario extra-escolar para practicar y ampliar sus conocimientos de idiomas extranjeros es Internet. Gracias a esta tecnología, el usuario puede encontrar una amplia información y emplear múltiples utilidades beneficiosas para el estudio. Al sistema de enseñanza o aprendizaje que utiliza las tecnologías proporcionadas por Internet se le conoce con el nombre de *e-Learning* (aprendizaje electrónico). Gracias a esta tecnología, los profesores pueden gestionar cursos *online* sin estar sometidos a una situación geográfica ni horario específico. Los participantes se conectan a los cursos mediante un dispositivo con conexión a Internet o conexión a una red privada (intranet) donde se hospede el curso.

*E-Learning* presenta una serie de pros y contras que deben tenerse en cuenta. Entre las ventajas que presenta, podemos destacar las siguientes:

- Gracias a Internet, se puede ofrecer a los alumnos una gran diversidad de material informativo.
- Se facilita la actualización, puesta al día y mantenimiento del curso.
- Permite el abaratamiento de costes para obtener información.
- Facilita al estudiante poder participar cuando lo vea posible.
- Facilita una formación *just in time* (en el momento que se necesite) y *just for me* (la formación se enfoca en cada alumno por separado).
- Ofrece varias utilidades de comunicación tanto sincrónica (*chat*, videoconferencia) como asincrónica (correo electrónico, foros de discusión) que facilita la comunicación profesor-alumno y entre alumnos.
- Favorece el aprendizaje mediante elementos multimedia.
- Facilita una formación en grupo y colaborativa.
- Ofrece facilidades para mantener un seguimiento de actividades realizadas por los estudiantes mediante la grabación de registros en el servidor.
- Ahorra costos y desplazamientos.

Por contra, nos encontramos con algunos inconvenientes a tener en cuenta:

- Depende de una conexión a Internet, cuanto más rápida mejor.
- El profesor necesita invertir más tiempo para atender a sus alumnos que por la forma tradicional.
- Requiere que los participantes tengan las nociones tecnológicas necesarias para saber dar uso al aplicativo correctamente.
- Requiere que los estudiantes sepan estudiar de forma autónoma.
- La calidad del aprendizaje se puede deteriorar con el tiempo si no se sigue el curso de forma regular.
- Requiere más trabajo de lo normal.
- Aún predominan los cursos asistenciales y se desconfía de esta modalidad de enseñanza.
- Los profesores no están formados para enseñar con esta metodología.
- No existe mucha experiencia con esta metodología.
- Aún existe una brecha digital: Hay población que socio-económicamente no se puede permitir la tecnología necesaria para poder acceder a Internet.

Hay que tener en cuenta estos inconvenientes porque son el producto de más del 80% de fracasos de los cursos a distancia y de más de un 60% de abandono por parte de los estudiantes. Algunos de estos inconvenientes se pueden solventar a lo largo del tiempo, mientras se va ganando experiencia en el uso del aplicativo y mediante una buena metodología de estudio.

Para poder realizar cursos en línea con un buen criterio de calidad, Rena Pallof y Keith Pratt [1] nos proponen una serie de “lecciones” o principios para una buena realización de un curso a distancia:

- *Principio 1: La buena práctica anima al estudiante a tomar contacto con la facultad.* El profesor debe ofrecer guías claras para poder interactuar con los estudiantes.
- *Principio 2: La buena práctica anima la cooperación entre los estudiantes.* Una discusión bien diseñada facilita significativamente la cooperación y participación de los estudiantes.
- *Principio 3: La buena práctica facilita un aprendizaje activo.* El estudiante debe realizar tareas y presentar proyectos a lo largo del curso.
- *Principio 4: La buena práctica implica un «feedback» rápido.* Es necesario que el profesor realice dos tipos de *feedback* (respuesta a un correo electrónico enviado): de información y de acuse (de haber recibido el correo).
- *Principio 5: La buena práctica pone énfasis en el tiempo de la tarea.* Los cursos a distancia requieren una fecha límite.
- *Principio 6: La buena práctica comunica elevadas expectativas.* El profesor debe provocar tareas, ejemplos de caso y alabanzas comunicando la calidad de los trabajos de sus estudiantes.
- *Principio 7: Las buenas prácticas respetan los diversos talentos y caminos de aprendizaje.* El profesor debe permitir a los estudiantes elegir el tema de los proyectos a realizar y dejar que surjan diferentes puntos de vista.

De acuerdo con Pallof [1], antes de comenzar un curso *online* deben plantearse las siguientes preguntas, básicas para el buen funcionamiento y comportamiento del curso:

- ¿Quiénes son los estudiantes?
- ¿Qué se quiere lograr por medio del curso? ¿Qué se desea que los estudiantes sepan, sientan o sean capaces de hacer como resultado del curso o experiencia? ¿Qué contenidos podrían soportar estos objetivos?
- En el curso, ¿qué tiene de exitoso transferir los contenidos a un contexto en línea?
- ¿Qué guías, reglas, funciones y normas se necesitan establecer para la culminación del curso?
- ¿Cómo planificar la distribución del curso? ¿Qué pueden esperar los estudiantes en el proceso de aprendizaje? ¿Qué puede ofrecer una combinación de las opciones de una enseñanza en línea y una presencial?
- ¿Cuán cómodo me encuentro como instructor con el aprendizaje colaborativo, la interacción personal, promoviendo el conocimiento con los estudiantes y liberando el control del aprendizaje?
- ¿Cómo se quiere organizar el sitio del curso? ¿Cuán flexible se quiere que sea para hacer cosas? ¿Se tiene la libertad de elegir el curso de la manera que se desee?
- ¿Cómo acceden los estudiantes a las sesiones del curso?
- ¿Cómo querer dirigirse a los requisitos de asistencia?
- ¿Cómo definir el aprendizaje y qué tener que ver como resultados de aprendizaje para la clase?

En el siguiente apartado se describe un análisis de diferentes LCMS existentes en el mercado y cuál es el idóneo para la realización del proyecto.

## **2.2 Análisis de recursos**

Teniendo en cuenta las premisas anteriores para la realización del proyecto, se llevó a cabo un estudio sobre diferentes sistemas LCMS existentes en el mercado actual. A continuación se introduce al lector al mundo de los LMS (*Learning Management System* – Sistema de Gestión de Aprendizaje). Más adelante se describen los LCMS más comunes y, finalmente, se argumenta cuál es el sistema idóneo escogido para la realización del proyecto.

### **2.2.1 LMS (*Learning Management System*)**

Con el panorama planteado en el estado del arte aparecen a la luz los *Sistemas de Gestión de Aprendizaje* (LMS). Un LMS es una aplicación *software* instalado en un servidor, utilizado para administrar, distribuir y controlar las actividades del *e-Learning* de una institución u organización. Principalmente, se encarga de planificar, repartir y manipular eventos de aprendizaje dentro de una organización, incluyendo cursos *online*, aulas virtuales y cursos tutorizados. Se enfoca en administrar usuarios, realizando un seguimiento de los usuarios a través de todas las actividades que puede ofrecer. También realiza tareas administrativas, como generar informes para los profesores.

Pero un LMS por sí solo no genera contenido de los cursos. Para generar el contenido se requiere un generador de contenidos para cursos conocido con las siglas LCMS (*Learning Content Management System*), que proporciona el medio necesario para crear y reutilizar contenido de *e-Learning* y reducir esfuerzos de desarrollar duplicados innecesarios. Sus principales metas son las siguientes:

- Control centralizado del contenido de aprendizaje de las organizaciones para una búsqueda eficiente y recuperación del contenido.
- La productividad gana alrededor de rápidas y condensadas cronologías de desarrollo.
- La productividad gana alrededor de ensamblajes, mantenimientos y publicaciones / técnicas de marca / entregas de contenido de aprendizaje.

Muchas empresas se han lanzado a producir distintas aplicaciones LCMS, algunas de pago, otras de libre distribución. Se presentan ahora una selección de los programas más relevantes en el mercado actual.

### **2.2.1.1 Sistemas LCMS de pago**

Estos sistemas, al ser de pago, carecen de la posibilidad de ser adaptados, modificados o ampliados por un desarrollador externo, lo que limita su funcionalidad a lo que llegue a ofrecer el sistema. Los sistemas LCMS que más destacan en este terreno son *Blackboard Learning System*, *WebCT* (aunque ahora está fusionado con *Blackboard*) y *Desire2Learn*.

*Blackboard Learning System* [2] es un producto de la empresa *Blackboard Inc.*, creado en 1997. Ofrece, entre otras cosas, la capacidad de administrar cursos, una arquitectura abierta adaptable y un diseño escalable que ayuda a la integración con sistemas de información de estudiantes y protocolos de autenticidad. Puede ser instalado en servidores locales u hospedado por soluciones *ASP* (*Active Server Pages*) de *Blackboard*. Su principal propósito es añadir elementos *online* emulando cursos tradicionalmente subministrados cara a cara, y desarrollar cursos completamente virtuales con pocas o ninguna reunión cara a cara. En el momento de esta escritura, la versión más actualizada de este sistema se llama *Blackboard Learn, Release 9*. Su instalación completa puede rondar los 250.000 dólares para soportar un servicio de hasta 25.000 estudiantes, un precio desorbitado teniendo en cuenta lo que nos ofrecen otros productos.

*WebCT* (*Web Course Tools*, o Herramientas para Cursos Web) [3] es un LCMS originalmente desarrollado en la Universidad de Columbia Británica, Canadá, por Murray Goldberg en 1995. En 1997, Murray creó la compañía *WebCT Educational Technologies Corporation*, empresa derivada de la universidad anteriormente mencionada. A mediados de 1999, *WebCT* fue adquirida por la empresa *ULT* (*Universal Learning Technology*). A finales del 2005, *Blackboard Inc.* y *WebCT* firmaron un acuerdo de fusión que suscitó miedos entre los expertos de anti-monopolio del Departamento de Justicia de los Estados Unidos. La fusión se realizó con el cambio de nombre a *Blackboard*. La fusión se terminó a comienzos del 2006. *WebCT* desarrolla dos versiones: *Vista*, desarrollado para empresas, y *Campus Edition*, desarrollado para universidades. No tiene costos establecidos, ya que depende del tamaño de la institución y de cómo va a ser usada. Su última versión, antes de fusionarse con *Blackboard*, presentaba muchos problemas por hacer uso indebido del control del usuario: Deshabilitaba el botón de retroceso del navegador de acceso a Internet, usaba Javascript con las URLs (*Uniform Resource Locator*) y su sistema de *tabs* (pestañas) era lo suficientemente confuso para pasar desapercibidos en según que situaciones, entre otros problemas de usabilidad.

*Desire2Learn Inc.* [4] es una empresa fundada en 1999 destinada al desarrollo de LCMS para escuelas, institutos, empresas, asociaciones o el gobierno. Su producto estrella se llama *eLearning Enterprise Suite* y comprende una serie de utilidades que abarcan todas las necesidades para implementar un LCMS muy funcional. El hecho de que sea de pago hizo que se descartara este producto. Además, hoy en día está en juicio a causa de supuestas violaciones de patentes del producto *Blackboard*.

*Dokeos* [5] es una entidad creada entre 1998 y 1999 por Thomas de Praetere en base al análisis de

otras herramientas existentes en la época para construir una alternativa mejor. Su meta principal es ser un sistema flexible y con gran facilidad de uso por parte de los usuarios. Cuenta con tres versiones, una gratuita, que comentaremos más adelante y dos de pago, *Dokeos Pro* y *Dokeos Medical*. Ambos productos son una versión mejorada y ampliada de lo que ofrecen con la versión gratuita, bastante limitada, destacando la versión *Medical*, un producto destinado a la enseñanza de conocimientos médicos, con herramientas específicas como gestor de casos de uso médicos.

### 2.2.1.2 Sistemas LCMS Open-Source

A diferencia de los LCMS de pago mostrados anteriormente, estos sistemas LCMS gratuitos *open-source* ofrecen, entre otras cosas, el código fuente del producto liberado para permitir la modificación y actualización por parte de un desarrollador externo, y se ofrecen de forma gratuita, dejando al cliente el único requerimiento de poseer un servidor adaptado para poder alojar el sistema. Entre los disponibles, los que más destacan en el mercado actual son *ATutor*, *Dokeos*, *Sakai Project* y *Moodle*.

*ATutor* [6] es un LCMS de código abierto cuyo objetivo principal es la accesibilidad y la adaptabilidad. Está bajo licencia *GNU GPL* (acrónimo recursivo que se lee *GNU's Not Unix General Public License*), licencia de software gratuito. *ATutor* comenzó como un producto de la entidad *Adaptive Technology Resource Center* de la Universidad de Toronto en el año 2003, pero hoy en día es mantenido y actualizado por programadores y gente que aporta ideas o comenta errores, sin ánimo de lucro. *ATutor*, además, es el primer LCMS en cumplir las especificaciones de accesibilidad *W3C WCAG 1.0* (*World Wide Web Consortium Web Content Accessibility Guidelines*) de nivel AA+. Estas normas especifican la metodología a seguir para realizar páginas web utilizables para personas con todo tipo de capacidades o discapacidades. La versión actual del producto (1.6.2), exige los siguientes requisitos para su uso:

- Servidor web HTTP (*HiperText Transfer Protocol*), preferiblemente que sea *Apache v2*.
- *PHP* como lenguaje configurado en el servidor web HTTP, en sus versiones 4.3.0+ o 5.0.2+, pre-configurado con *Zlib*, una biblioteca de compresión de datos; y la extensión *MBString* (*MultiByte String*), que ofrece funciones para aquellos idiomas que requieren un mapeado de caracteres mayor de 8 bits, como el japonés.
- *MySQL v4.1.0+* como base de datos. Además, se tendrá que instalar la extensión *MySQL* en *PHP* para poder interactuar con el lenguaje.
- Los navegadores deben soportar algunas de las técnicas avanzadas que utiliza, por lo que se requieren los siguientes navegadores para un funcionamiento completo de la aplicación: *Mozilla Firefox 2+*, *Opera 8+* y *Microsoft Internet Explorer 7+*. No obstante, se podrá seguir utilizando navegadores más antiguos para el funcionamiento básico.

*Dokeos Free* es el LCMS de código abierto que nos ofrece *Dokeos* [5] de forma gratuita. Está bajo licencia *GNU GPL*. En la actualidad, se ofrece la versión 1.8.6 desde su página web. Esta versión exige los siguientes requisitos para su uso:

- Servidor web HTTP, preferiblemente *Apache v2*.
- *PHP v5.0.2+* como lenguaje configurado en el servidor web HTTP.
- *MySQL v4.1.0+* como base de datos. Además, se tendrá que instalar la extensión *MySQL* en *PHP* para poder interactuar con el lenguaje.

*Sakai Project* [7] es otro LCMS de código abierto cuyo objetivo es ayudar a instructores, investigadores y estudiantes a colaborar *online* a realizar sus trabajos, tanto si es realizando cursos como investigaciones o colaboraciones de proyectos generales. Está bajo licencia *ECL v2.0*

(*Educational Community License*), modificación de la licencia ofrecida para el *Apache* v2.0 para cambiar el ámbito de la concesión de la patente para ser específica a las necesidades de las comunidades educacionales usando esta licencia. *Sakai Project* comenzó en el año 2004 cuando universidades como Stanford, Michigan, Indiana, MIT (*Massachusetts Institute of Technology*) y Berkeley comenzaron a crear un LMS común en vez de continuar sus propios sistemas o comprar software educacional a empresas externas. La *Mellon Foundation* proporcionó el capital necesario para la realización del proyecto. Desde entonces, como otros LCMS gratuitos, ha ido creciendo y mejorando gracias a la colaboración de entidades externas y usuarios sin ánimo de lucro. En la actualidad se ofrece la versión 2.5.4 desde su página web. Esta versión exige los siguientes requisitos para su uso:

- Servidor web HTTP, preferiblemente *Apache Tomcat* v5.5.26.
- *Java J2SE JDK* v5.0 Update 17+ (*Java 2 Platform Standard Edition Java Development Kit*), requerido para compilar el código de *Sakai Project*.
- *MySQL* v4.1.12+ o v5.0.45+, o *Oracle* 10g o 9i, como base de datos. Para poder interactuar con *Java J2SE JDK*, dependiendo de la base de datos elegida, se deberán instalar unos controladores para java específicos.

*Moodle (Modular Object-Oriented Dynamic Learning Enviroment)* [8] es otro LCMS de código abierto cuya meta principal es ayudar a los instructores para crear cursos *online* con muchas oportunidades para una interacción enriquecedora. Está bajo licencia *GNU GPL*. La versión 1.0 de *Moodle* fué creada por Martin Dougiamas, antiguo administrador de *WebCT* en la Universidad Tecnológica de Curtin, a mediados del 2002 aplicando las ideas del constructivismo en pedagogía que afirman que el conocimiento se construye en la mente del estudiante en lugar de ser transmitido sin cambios a partir de libros o enseñanzas, y del aprendizaje colaborativo. Desde entonces, ha recibido centenares de colaboraciones externas tanto por parte de entidades como por desarrolladores sin ánimo de lucro. A día de hoy, *Moodle* es el LCMS más utilizado del mundo, con más de 40.000 sitios registrados que incluyen en total más de 2.500.000 de cursos, entre los cuales se incluye la Universidad Autónoma de Bellaterra, ofreciendo la posibilidad de hospedar cursos *Moodle* a través de la herramienta *Campus Virtual*. Su última versión, v1.9.4+, ofrecida a través de su página web, debe cumplir los siguientes requisitos para su instalación:

- Servidor web HTTP, preferiblemente *Apache* v2 o *IIS (Internet Information Server)*.
- *PHP* como lenguaje instalado en sus versión 4.3.0+ o versiones más modernas. Se recomienda que éste tenga las extensiones pre-instaladas para la mejora de calidad del servicio.
- Se recomienda como base de datos el uso de *MySQL* o *PostgreSQL (SQL* corresponde a las siglas *Structured Query Language*), aunque gracias a la fusión con la librería *ADODB (Active Data Objects DataBase)*, puede gestionar más tipos, como *Microsoft SQL Server 2005*. *PHP* tendrá que estar preparado con las respectivas extensiones dependiendo de la base de datos utilizada.

### 2.2.1.3 Conclusión

Para la realización del proyecto, *ATutor* cumple muchas de nuestras necesidades, pero la falta de documentación clara para el desarrollo de nuevos módulos y la modificación del propio código hizo considerarlo como alternativa secundaria. *Dokeos Free* es un sistema bastante limitado en comparación a su hermanos mayores y no ofrece muchas de las funciones que podemos obtener con productos como *Moodle*. *Sakai Project*, al ser para una plataforma Java, fue descartado directamente, ya que interesa que se trabaje sobre PHP, que es la plataforma utilizada en los



espacios web de la UAB. *Moodle* es la herramienta que se ha elegido por ser el producto que ya utiliza la UAB para cursos *online*, y la que mejor documentación proporciona a nivel de desarrollador, aunque dista mucho de ser completa.

## 2.3 Planificación y organización de tareas

Durante la planificación inicial del proyecto se planteó la siguiente organización temporal de tareas. La primera parte consistía en la preparación de una plataforma *Moodle* con lo más básico del proyecto (foro, control de grupos, control de profesores y alumnos), para que los estudiantes y profesores pudieran trabajar sobre ella durante el segundo semestre. El tiempo estimado para esta primera parte se desglosó en las siguientes etapas:

- Estudio y configuración de *Moodle* en estado básico (sin modificación de código ni instalación de aplicaciones externas). **Tiempo:** 3 semanas.
- Petición de acceso al código fuente de *Moodle*. **Tiempo:** 1 semana.
- Instalación de *plug-in* externos ya creados. **Tiempo:** 3 semanas.
- Corregir problemas que se planteen tras la configuración del *Moodle* y la instalación de *plug-in*. **Tiempo indefinido.**
- Poner en funcionamiento el aplicativo en manos del profesorado para que puedan utilizarlo. **Tiempo:** 2 semanas.

La segunda parte consistía en la creación de extensiones al foro y solventar problemas ocasionados durante la primera etapa. Dichas extensiones serían asignadas a *Moodle* para que los alumnos y profesores fueran haciendo pruebas con ellos. El tiempo estimado para esta segunda parte se desglosó en las siguientes etapas:

- Diseño de los *plug-in*. **Tiempo:** 2 meses.
- Crear *plug-in* externo de reconocimiento de caracteres japoneses. **Tiempo:** 4 meses.
- Solventar problemas generados con las pruebas realizadas por los profesores y alumnos. **Tiempo indefinido.**
- Crear *plug-in* externo para el aprendizaje y memorización de vocabulario. **Tiempo:** 4 meses.
- Crear manual de utilización. **Tiempo:** 3 semanas.

Se previó que algunas de estas tareas se superpondrían. La memoria se iría escribiendo a lo largo del tiempo de duración del proyecto.

Durante el proceso de la primera parte hubo muchos contratiempos con la instalación de *Moodle*. El ordenador de pruebas tenía instalado *Windows Vista* y *Moodle* requiere de permisos especiales que el sistema operativo bloqueaba. Fue necesario pedir una máquina en la universidad para poder instalarlo temporalmente, y la instalación tardó 2 semanas más de lo previsto.

También hubo un contratiempo a la hora de administrar el aplicativo al profesorado. Tras la modificación de código interno de *Moodle*, se exigió un año de pruebas en una máquina facilitada por los técnicos de la UAB antes de poderse instalar la plataforma en sus servidores. Para solucionar este problema, se realizó un análisis para separar lo manipulado a un módulo externo, creando un duplicado del foro, pero *Moodle* estaba muy arraigado al módulo del foro nativo y solicitaba siempre una instancia de este foro, por lo que se tuvo que descartar la idea. Esto alargó el tiempo de trabajo otras 2 semanas y paralizó la participación durante la realización del proyecto de los usuarios hasta que el servicio técnico de la UAB diera permiso a utilizar la plataforma creada a

los estudiantes, una vez terminado el proyecto.

La segunda etapa produjo una cantidad mayor de contratiempos, a causa de la mala documentación sobre el desarrollo de los módulos. Si es cierto que hay una buena documentación de diseño básico, también es cierto que hay una falta de documentación a nivel más detallado de las diferentes funcionalidades que ofrecen los objetos de *Moodle*, obligando durante un largo periodo de tiempo a realizar pruebas hasta conocer qué funcionalidades ofrecía. Estas pruebas se alargaron durante 5 meses, haciendo imprescindible el rediseño del contenido de los bloques, obligando a descartar la creación de los *plug-in* de aprendizaje de vocabulario, para centrar el trabajo en el diseño de una buena utilidad de aprendizaje de kanji.

El siguiente capítulo describe los requisitos que requiere *Moodle* para funcionar correctamente, así como las diferentes herramientas utilizadas para el desarrollo del proyecto.

## 3. Plataforma y herramientas de desarrollo

En este capítulo se describe la plataforma *Moodle* y las herramientas utilizadas para el desarrollo del proyecto.

### 3.1 Plataforma Moodle

La versión empleada de *Moodle* (v.1.9.2), utiliza una arquitectura fuertemente orientada a objetos, aunque no por ello descarta el uso de la programación imperativa. Encontramos una estructura modular basada en la interconexión de módulos modificables (*plug-ins*). *Moodle*, además, permite el desarrollo externo de nuevos módulos que agregarán a la plataforma nuevas funcionalidades para los usuarios.

Para el uso correcto de *Moodle*, son necesarios unos requisitos tanto a nivel de servidor como a nivel de usuario. Se describen a continuación.

#### 3.1.1 Requisitos de servidor

La instalación correcta de la aplicación requiere toda una serie de elementos y utilidades externos.

##### 3.1.1.1 Disco duro

*Moodle* necesita un servidor web HTTP como *Apache* (recomendado por Moodle [9]) o *IIS*. También requiere de un intérprete de *PHP* [10], vinculado al servidor web, que permita traducir los *scripts* de *Moodle* al navegador. Para el funcionamiento correcto de *Moodle*, además, se recomienda la instalación extensiones para *PHP* destinadas a la gestión de código *Unicode* y otras mejoras. Por suerte los propios intérpretes oficiales ya vienen con los recomendados.

Para almacenar todo tipo de registros, se requiere una base de datos instalada en el servidor. Gracias a la integración en *Moodle* de la librería *ADODB* (*Active Data Objects DataBase*) [11], encargada de abstraer el acceso a múltiples bases de datos para *PHP*, *Moodle* puede manejar el uso de cuatro bases de datos: *MySQL* (v.4.1.12+), *PostgreSQL* (v.8.0+), *MS SQL 2005* (v.9 o v. *Express Edition*) y *Oracle* (v.10g+) [42]. La recomendada por *Moodle* es *MySQL*.

Existen productos preparados por *Moodle* que se encargan de realizar una instalación de servidor optimizada para varios sistemas, con solamente lo básico de lo que requiere *Moodle* de cada aplicación. Para ello, usa la aplicación *XAMPP* (*X, Apache, MySQL, PHP, Perl* [la X simboliza el soporte multiplataforma]) [13], un paquete de servicio web multiplataforma que contiene la instalación básica de *Apache*, *MySQL*, *PHP* y *Perl*, otro lenguaje de *scripts*. La versión para *Windows*, descargable desde la web oficial de *Moodle*, requiere menos de 200 MB de disco duro para poder ejecutarse. Este paquete ha sido el utilizado para el ordenador de pruebas del proyecto sobre un sistema operativo *Windows Vista x64 Ultimate*. Con *Linux* existe una alternativa, sobretodo gracias al uso del sistema de gestión de paquetes *APT* (*Advanced Packaging Tool*) en sistemas *Debian*, que recuperan lo básico y necesario para la ejecución de *Moodle*, necesitando un espacio de disco parecido al obtenido mediante el paquete *XAMPP* de *Windows*.

Si se guarda material didáctico extra, se instalan nuevas extensiones o se contempla la participación de muchos usuarios que conlleve la utilización masiva de la base de datos, se exigirá mayor espacio de disco.

Para este proyecto, se necesita espacio para almacenar diccionarios de japonés para kanji y vocabulario (ver Anexo 1), módulos de aprendizaje de japonés, compuesto por tres bloques que

ocupan 53 MB (contienen internamente los diccionarios) más un módulo de actividades que ocupa poco menos de 200 KB. También se debe tener en cuenta que los profesores ofrecen la posibilidad de subir fotografías u otros archivos multimedia por los alumnos, siendo recomendable que el disco duro del servidor ofrecido para el proyecto tenga 10 GB de memoria, para evitar problemas a corto plazo.

### 3.1.1.2 Memoria RAM

La memoria RAM se encarga de soportar un cierto número de usuarios en línea concurrentemente. Cuanta más memoria RAM instalada en el servidor, mayor número de usuarios podrá soportar a la vez. Desde la web oficial de *Moodle* [8] se recomienda como mínimo que el servidor posea 256 MB de memoria RAM, aunque es preferible 1 GB (*GigaByte*) de memoria RAM.

Para la correcta elección de la cantidad de memoria RAM necesaria para un servidor, de forma general, se tienen que tener en cuenta dos parámetros:

1. **Usuarios que navegan por Moodle:** Es importante conocer la cantidad máxima de usuarios que pueden conectarse a la plataforma. Normalmente esta cantidad es el número de ordenadores que pueden acceder al servidor.
2. **Usuarios que acceden a la base de datos concurrentemente:** Es importante conocer el número de usuarios esperados a conectarse simultáneamente a la base de datos, pues ésta tiene configurado un límite de usuarios concurrentes, y *Moodle* es propicio a llegar a este máximo teniendo en cuenta la posibilidad de realizar actividades tales como exámenes.

Con estos parámetros y ayudado de una buena elección de la CPU (*Central Processing Unit – Unidad Central de Procesamiento*), junto con el tipo de red donde se instale el servidor y el número de servidores a utilizar, se puede obtener una estimación cercana, aunque no perfecta, del mínimo de memoria RAM necesaria para el servidor. La regla general para calcular la memoria RAM necesaria para un sólo servidor viene dada por el siguiente cálculo:

$$\text{Máximo de usuarios concurrentes DB} \simeq X \text{ GB RAM} \times 50$$

El máximo de usuarios que navegan por *Moodle* viene dado por este otro cálculo:

$$\text{Máximo de usuarios en Moodle} \simeq \text{Máximo de usuarios concurrentes DB} \times 5$$

Como objetivo del proyecto se plantea hacer un curso de entre 10 y 15 grupos con menos de 10 alumnos por grupo, lo que nos da una media de 125 participantes. Teniendo en cuenta que una tercera parte de los alumnos son residentes en Japón y el resto en España, la diferencia horaria entre ambos países podría dificultar la concurrencia. Si además tenemos en cuenta que la participación en años anteriores ha sido relativamente baja, se puede concretar que con 1 GB de memoria RAM (según los cálculos anteriores, una media de 250 usuarios en línea a la vez) la plataforma *Moodle* tendrá suficiente para aguantar el sistema correctamente. No debemos olvidar que el proyecto pretende ser instalado dentro de la red *Moodle* alojada en los servidores de la UAB, por lo que la fluidez del curso vendrá dado por lo que tenga que soportar el servidor a lo largo del curso.

### 3.1.1.3 CPU

Cuanto mejor sea la CPU, más rápido se podrán interpretar las páginas y enviar pronto los datos al cliente, pero sin la ayuda de una buena red de conexión o la memoria RAM mínima necesaria, el tener mucha CPU no solventará muchos problemas. Aún así, la CPU no es el elemento más importante a tener en cuenta ya que hoy en día los procesadores ofrecen bastante buen rendimiento.

#### **3.1.1.4 Red de conexión**

Cuanto mejor sea la red, menos tiempo de transferencia de datos y, en este tipo de aplicaciones donde se mueven muchos documentos tanto de profesor a alumno como de alumno a profesor, una mala instalación de red puede causar efectos muy negativos a nivel de usuario que puede terminar con la pérdida de interés por el curso. Fuera de la red donde esté vinculado el servidor, dependerá también de la conexión que pueda permitirse el usuario.

### **3.1.2 Requisitos de los usuarios**

*Moodle* requiere que el dispositivo del usuario sea capaz de conectarse a la red donde esté instalado el servidor (tanto intranet como Internet) y que disponga de un navegador capaz de interpretar los datos ofrecidos por el servidor. Se recomienda que el navegador utilizado sea capaz de manejar *cookies*, ejecutar código *Javascript* e interpretar correctamente las hojas de estilo CSS (*Cue-Style Sheet*). Los navegadores recomendados son *Mozilla Firefox* v2.X+ [15], *Microsoft Internet Explorer* v6.0+ [16], *Safari* v3+ [17] y *Google Chrome* v1.X+ [18]. Para el proyecto, además, es requerido la instalación del *plug-in* de *Adobe Flash Player* [43] para poder hacer uso de la pizarra de entrenamiento.

### **3.2 Herramientas de desarrollo**

Para el desarrollo del proyecto se ha dispuesto del siguiente material:

- Un portátil con las siguientes características:
  - *AMD Turion64* x2 de 2 Ghz.
  - 2 GB de memoria RAM.
  - Partición del disco duro de 168 GB.
  - Sistema operativo *Windows Vista Ultimate x64*.
- Un ordenador-servidor con sistema Unix instalado para las pruebas facilitado por el Servicio Técnico del Departamento de Arquitectura de Computadores y Sistemas Operativos (DACSO-CAOS) de la UAB.

Para la instalación de la plataforma y el desarrollo de las modificaciones y extensiones necesarias se emplean aplicaciones que se comentan a continuación.

#### **3.2.1 XAMPP**

*XAMPP* [13] es un paquete de utilidades multiplataforma gratuito que ayuda a convertir una maquina en un servidor web de forma fácil y controlada. Los programas que contiene son los siguientes: El servidor web *Apache*, los intérpretes *PHP* y *Perl*, y la base de datos *MySQL*, además de otras pequeñas utilidades y extensiones preparados para los diferentes sistemas operativos que puede instalarse (*Windows*, *Linux*, *Mac OS X* y *Solaris*, entre otras variantes).

La versión elegida para el desarrollo del proyecto ha sido un paquete preparado por la propia página oficial de *Moodle* para *Windows* que viene con todo pre-configurado, sólo es necesario descomprimir en una carpeta del sistema y ya está listo para ser utilizado.

#### **3.2.2 Aptana Studio**

*Aptana Studio* [19] es un entorno de código libre destinado al desarrollo de aplicaciones web con

*AJAX (Asynchronous JavaScript And Xml)*. *Aptana Studio* está desarrollado por *Aptana Inc.*, y comenzó en 2006 con el nombre de *Aptana IDE* como un *plug-in* para *Eclipse*, otro entorno de desarrollo gratuito de código libre especializado para la programación en *Java*. En 2007 *Aptana Inc.* ofreció una versión del *plug-in* como programa autónomo pasándose a llamar *Aptana Studio 1.0*, basado en la interficie de *Eclipse*. Hoy en día se ofrece tanto la versión autónoma como la versión *plug-in* para *Eclipse*. Su versión más actual, la versión 1.5.1, soporta la programación en *PHP*, *Ruby on Rails*, *Python* y *Adobe AIR* mediante la instalación de módulos externos. Dispone además de un producto comercial (*Aptana Studio Pro*), con mejoras no disponibles para la versión gratuita. Para la realización del proyecto fue suficiente la versión gratuita y el módulo de *PHP*.

El proyecto de *Moodle* instalado se integró por completo como proyecto de *Aptana Studio*, pudiendo éste acceder a todas las librerías de *Moodle* para un desarrollo más fructífero.

### 3.2.4 jQuery

*jQuery* [20] es un *framework* escrito en *Javascript* gratuito y *open-source* bajo doble licencia (*MIT License* y *GNU General Public License*), desarrollado por John Resig en 2006. Ayuda al desarrollador en tareas de manipulación del árbol *DOM (Document Object Model)*, control de eventos, efectos visuales y en el uso simple de la tecnología *AJAX*. La versión actual y utilizada (v1.3.2) es ampliable mediante *plug-ins* externos. Para el proyecto se ha hecho uso de los siguientes *plug-ins*:

- ***jCarousel*** [21] es un *plug-in* creado por Jan Sorgalla en 2006, cuya función es la de controlar un listado de elementos en orden vertical u horizontal. El contenido de los elementos puede ser tanto HTML estático como dinámico, generado a partir de la obtención de datos mediante *AJAX*. Además, ofrece funcionalidades tanto de desplazamiento manual como automático del listado, mostrando un número de elementos limitado del listado total. A este tipo de control se le conoce como *carrusel*, ya que ofrece la sensación de ver un carrusel dando vueltas. Es gratuito y tiene doble licencia tanto *GPL* como *MIT*. La versión utilizada es la 0.2.3.
- ***jTemplates*** [22] es un *plug-in* gratuito creado a mediados del año 2007 por una persona que se hace llamar *tpython*, y tiene doble licencia *open-source* bajo *MIT* y bajo *GPL*. Es un potente motor de plantillas para *Javascript*, útiles para la creación dinámica de contenido desde el navegador cliente, sin requerir el uso del servidor. La versión utilizada es la 0.7.8.
- ***blockUI*** [23] es un *plug-in* gratuito bajo doble licencia *MIT* y *GPL* creado por Mike Alsup en 2007. Se encarga de *simular el comportamiento sincronizado en llamadas asíncronas mediante AJAX sin bloquear el navegador*. Útil para reducir problemas entre llamadas asíncronas al servidor.

### 3.2.4 Adobe Flash

*Adobe Flash* [24] es una plataforma multimedia destinada a la vectorización y rasterización de imágenes y videos. Inicialmente fue creada con el nombre de *FutureSplash Animator* por la compañía *FutureWave* a mediados de 1996. A finales de 1995 *Macromedia* compra el programa, rebautizándolo como *Macromedia Flash* hasta que, a mediados del 2005, ésta anuncia un acuerdo de compra por parte de *Adobe Systems*, rebautizándolo como *Adobe Flash*. Actualmente se ofrece al mercado la versión *Adobe Flash CS4*, utilizada para el proyecto.

*Adobe Flash* utiliza un *script* para controlar dinámicamente sus contenidos llamado *ActionScript*, actualmente en su versión 3. Para el proyecto, en cambio, se utilizó su versión anterior (v2) por haber disponible más información y documentación de desarrollo.

En el siguiente capítulo se describen las diferentes partes del diseño y desarrollo del proyecto.

## 4. Diseño e implementación

Para cumplir los requisitos de los profesores, el desarrollo del proyecto se dividió en dos partes bien diferenciadas. La primera parte estuvo marcada por la preparación de la plataforma *Moodle* (instalación, configuración y búsqueda de *plug-ins* externos) y la modificación de código interno para mejorar el foro de *Moodle*. La segunda parte consistió en la ampliación de la estructura de base de datos y la creación de nuevos módulos para la plataforma.

### 4.1 Primera fase: Preparación de la plataforma Moodle

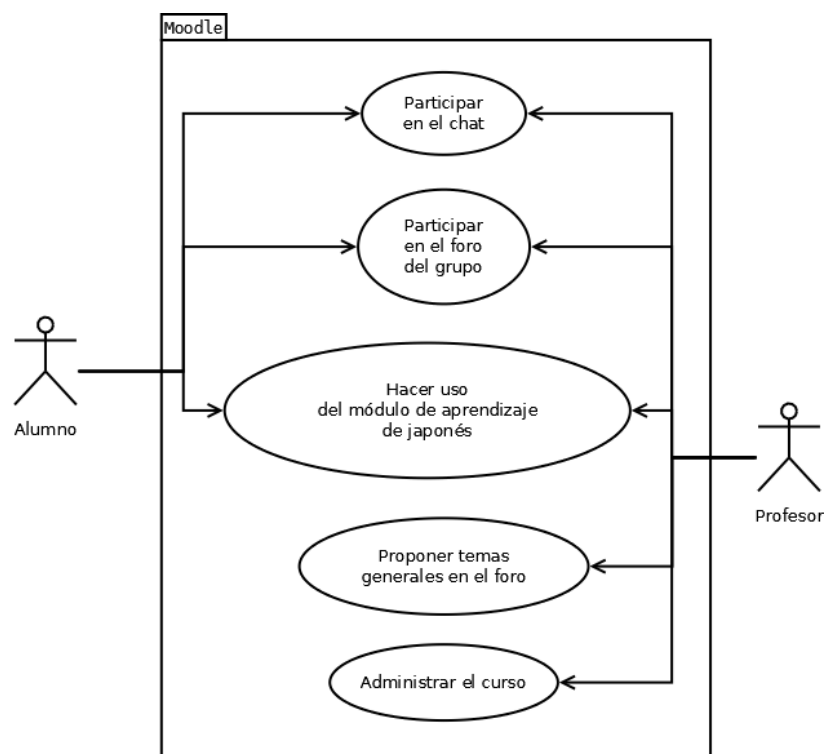
Al elegir la plataforma Moodle como medio de trabajo, se investigó hasta qué punto *Moodle* podía satisfacer los requerimientos solicitados (ver la figura 8 para ver un caso de uso). Una vez conseguido instalar la plataforma *Moodle* en el ordenador de desarrollo, comenzó la creación de un curso que adaptara las siguientes características:

1. El foro es el tema principal del curso, todo lo demás es secundario.
2. El curso tiene que poder dividir a los estudiantes en grupos pequeños de 5 o 6 personas.
3. Los grupos están organizados por los profesores a partir de la diferencia de conocimientos que ofrezcan sus alumnos.
4. Los alumnos de un grupo concreto no deben conocer lo que ocurre en los otros grupos.
5. Los profesores tienen acceso a todos los grupos y pueden participar en sus foros.
6. Los profesores tienen la posibilidad de escribir temas de tipo general que podrán ver todos los alumnos sin restricción de grupo. Útil para dar noticias y ejercicios de discusión.
7. Los usuarios, tanto profesores como alumnos, disponen de una sala de *chat* en la que entablan conversación *online*. También se organiza por grupos.
8. Es interesante el uso de la videoconferencia para el aprendizaje.
9. Es deseable que se muestre el curso tanto en español como en japonés como mínimo, inglés y catalán son optativos.

Los puntos del 1 al 5, el 7 y el 9 fueron fácilmente solventados por *Moodle*. Ofrece de serie la posibilidad de crear un tipo de curso fuertemente enfocado al foro conversacional (curso de tipo Social) y permite la configuración de un curso dividido por grupos de usuarios. Los profesores y los administradores pueden modificar los cursos agregando nuevos módulos, como el bloque de *chat*, o administrando los grupos de usuarios. Sobre el punto 9, *Moodle* dispone de un catálogo *online* de paquetes de idiomas que puede descargarse y filtrar el listado de idiomas a mostrar en el curso.

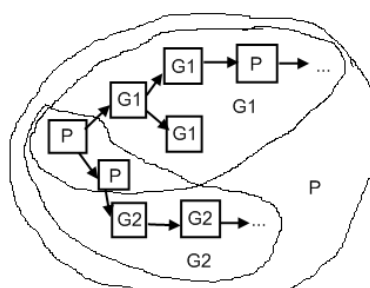
El punto 8 produjo muchos contratiempos. Se buscó la existencia de módulos destinados a la videoconferencia a partir del listado de *plug-ins* oficiales de Moodle [25]. El *plug-in* más completo fue *COVCELL Audio/Video Conferencing Tool* [26], creado por el proyecto *COVCELL (Cohort-Oriented Virtual Campus for Effective Language Learning)*, un proyecto que trabaja sobre el campo del aprendizaje de idiomas y la enseñanza online, y por el desarrollo de software abierto en educación [26]. Tuvo que desestimarse su utilización porque exigía un servidor Unix con JRE (*Java Runtime Enviroment*) v1.5+ instalado únicamente para el *streaming* de video, y no se disponía para el desarrollo del proyecto.





**Figura 8 – Caso de uso del curso**

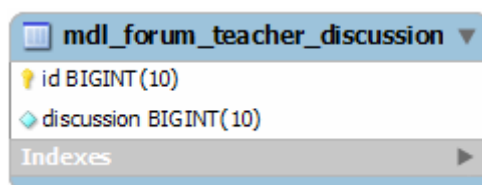
Para el punto 6 se tomaron decisiones que afectaron la finalidad del proyecto. *Moodle* no era capaz de permitir la creación de temas globales por parte de los profesores donde pudieran contestar los alumnos y que éstos mantuviesen el anonimato de la participación para los alumnos de otros grupos.



**Figura 9 – Esquema de un tema global modificado**

Un ejemplo lo podemos encontrar en la figura 9. El árbol representado es un tema global creado por un profesor (P – Nodo raíz), cuyos nodos de las ramas son las respuestas dadas tanto por los alumnos de los grupo G1 y G2 como por los mismos profesores. Se muestra con círculos el campo de visibilidad de cada grupo de alumnos y de los profesores.

Para solventar el problema tuvo que agregarse a la base de datos una nueva tabla (ver figura 10) que se encargara de identificar los temas globales con el permiso de participación de los estudiantes. Además, varios ficheros del módulo del foro fueron modificados para conseguir el anonimato de los grupos. Para más información sobre los cambios realizados a nivel de código, véase el anexo 3 de este documento. Uno de esos cambios fue agregar un campo nuevo al formulario de creación de temas para los profesores, oculto a los alumnos, como se puede ver en la figura 11. Este campo permite decidir si un tema global puede ser contestado por alumnos o no.



**Figura 10 – Tabla nueva para la mejora del foro**

**Figura 11 – Cambios en el formulario de creación de nuevos temas globales**

Terminadas las tareas, se decidió mejorar los servicios ofrecidos para hacer del proyecto algo más interesante que un simple foro de intercambio cultural. Se quiso dar una visión más adecuada al proyecto como lugar de aprendizaje de lenguas.

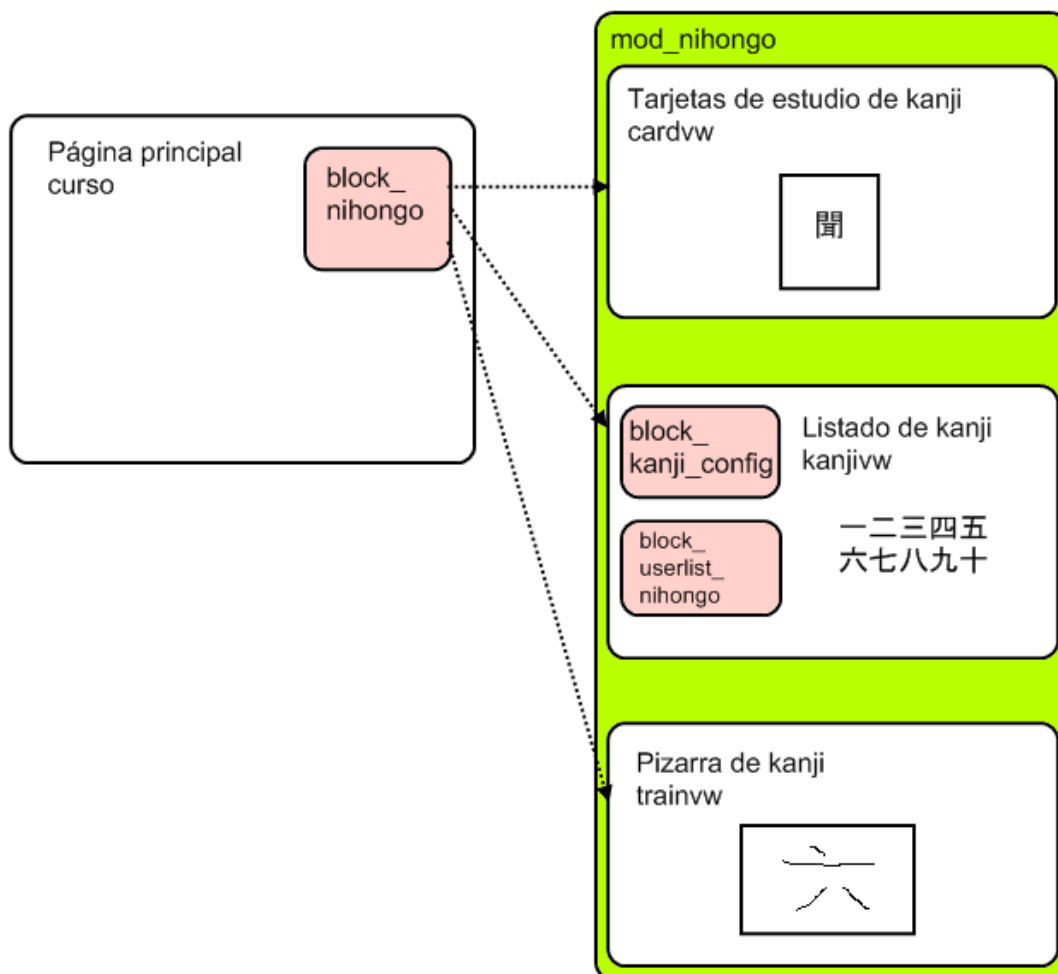
## **4.2 Segunda fase: Creación del módulo de aprendizaje de japonés**

El objetivo de esta segunda fase fue ofrecer a los alumnos la posibilidad de mejorar sus conocimientos sin la participación de los profesores. Para ello se propuso la utilización de diccionarios, fichas de aprendizaje de kanji y una pizarra para practicar la escritura de los kanji, con lo cual, tras una nueva toma de requisitos, se comenzó el desarrollo de un módulo de aprendizaje de caracteres japoneses para que pudieran hacer uso los alumnos.

Para conseguir nuestro objetivo, se desarrolló una nueva serie de extensiones interconectadas tal y como se puede ver en la figura 12:

- **block\_nihongo:** Un bloque insertado en la página principal del curso que da acceso a las diferentes actividades de aprendizaje de kanji ofrecidas por el módulo de actividades desarrollado. Contiene un control de servidor llamado *Palabra del Momento* que muestra, mediante la técnica de *carrusel*, un kanji y una palabra del vocabulario de forma aleatoria durante un tiempo especificado. Este bloque, además, configura el idioma del usuario con el que se traduce el significado del kanji y del vocabulario.
- **mod\_nihongo:** Un módulo de actividades para el aprendizaje de caracteres japoneses dividido en tres partes importantes

- **cardvw**: Un generador de tarjetas de estudio de kanji.
- **kanjivw**: Una página que muestra listados de kanji que maneja dos bloques para enriquecer la navegación del usuario:
  - **block\_kanji\_config**: Un bloque destinado a la configuración del listado de kanji, capaz de manejar tanto el modo de agrupación de kanji como el número de filas y columnas de kanji por página.
  - **block\_userlist\_nihongo**: Un bloque destinado al mantenimiento de listas predefinidas de kanji por usuario. Estas listas permiten almacenar kanji concretos, tantos como se deseen por lista.
- **trainvw**: Una pizarra para practicar la escritura de los números japoneses en forma de kanji.



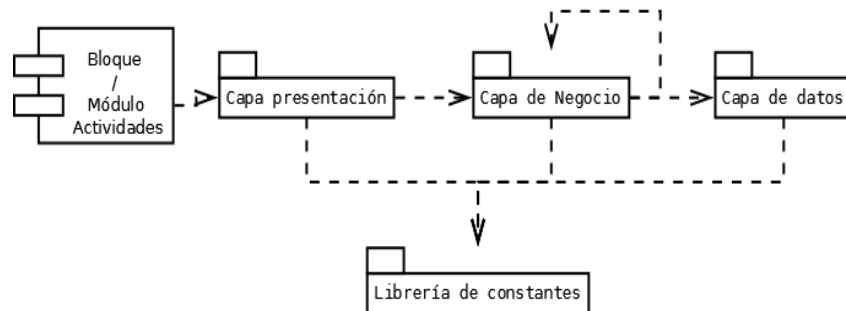
**Figura 12 – Interconexión de los módulos desarrollados**

Es importante la instalación de todos ellos en la aplicación para el correcto funcionamiento de los módulos.

#### 4.2.1 Arquitectura de los módulos

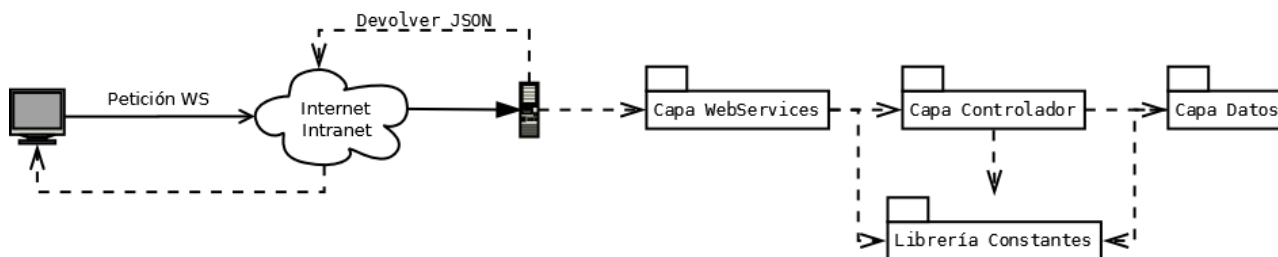
Tras una descripción de los módulos deseados, se siguieron unos patrones comunes de diseño, aún desobedeciendo las buenas practicas que se recomiendan para el desarrollo de los módulos de Moodle [44], pero que han demostrado ser eficientes.

- Los módulos deben usar una arquitectura multicapa (Ver figura 13). La arquitectura multicapa es *un estilo de programación en el que el objetivo primordial es separar la lógica de negocio de la lógica de diseño* [26]. Cada capa es un conjunto de clases, aplicando la programación por objetos. Todas las capas comparten acceso a una librería de constantes, un conjunto de constantes que abarca valores por defecto, código de funciones para Web Services o enlaces a archivos..



**Figura 13 - Estructura básica de la arquitectura multicapa de los módulos y bloques**

- La capa de presentación se encarga de gestionar la configuración de los módulos y de generar código *HTML* y *Javascript*. Tiene acceso a la capa de negocio. Los módulos son los únicos objetos que tienen acceso a esta capa.
- La capa de negocio se encarga de realizar el proceso de validación y manipulación de datos. Es capaz de acceder a otras clases de la capa de negocio y a la capa de datos.
- La capa de datos es la única capa con acceso a la base de datos.
- Moodle recomienda el uso del *framework YUI* de *Yahoo!*, pero tras investigar se decidió hacer uso de la librería *jQuery* por ser simple y potente [28]. Para el uso de *jQuery* se generó un directorio nuevo llamado *jQuery* dentro del directorio *lib* de *Moodle*, lugar donde se encuentran todas las librerías genéricas, además de las extensiones de terceras compañías. En esta carpeta se agregaron todos los *plug-in* utilizados por los módulos además del *framework*, y se creó una función de generación de *tags* de acceso a los *scripts* tanto del *framework* como de los *plug-ins* utilizados.
- A nivel de cliente se debe usar una arquitectura multicapa tal y como se muestra en la figura 14.
  - El navegador cliente debe poder ejecutar código *Javascript* para poder hacer uso de la tecnología *AJAX* (*Asynchronous JavaScript And Xml*). *AJAX* proporciona conexiones asíncronas al servidor mediante un *WS* (*Web Service*) para obtener solamente la información deseada, sin tener que recargar toda de la página. Las llamadas a *WS* localizados en el archivo *ws.php* se realizan mediante *POST* y tienen una serie de parámetros de entrada básicos para todas las llamadas:
    - **wsfunction**: Numérico entero. Indica el código de la función a ejecutar por el servidor.
    - **userid**: Numérico entero. Indica la id del usuario. Necesaria para validar si el usuario que solicita es el mismo que está conectado.



**Figura 14 - Arquitectura multicapa empleada para la conexión asíncrona de los módulos.**

- Cuando el usuario realiza una petición al WS, éste se encarga:
  - de recuperar los parámetros pasados por *POST* por el navegador cliente, validando su tipo,
  - de validar que el usuario que procesa la petición es el mismo que *Moodle* tiene controlado como usuario conectado,
  - y de llamar a la capa WS dependiendo de la funcionalidad deseada por el usuario.
- La capa de WS contiene un conjunto de clases enfocadas a la gestión de peticiones asincronas del usuario y a la generación de objetos *JSON* (*JavaScript Object Notation*). Como las capas de los módulos, tiene acceso a la librería de constantes y, como capa, solamente tiene acceso a la capa de negocio.
- Durante el proceso en el WS, si se produce algún error grave controlado, se devuelve un mensaje de error con código de respuesta HTTP 500 (*Internal Server Error*).
- Una vez procesada la petición, envía un objeto *JSON* de vuelta al cliente como mensaje HTTP de tipo de contenido (*content-type*) *application-json*. *JSON* ofrece una alternativa al *XML* (*eXtensible Markup Language*) porque son objetos *Javascript* y *jQuery* se encarga de su conversión.

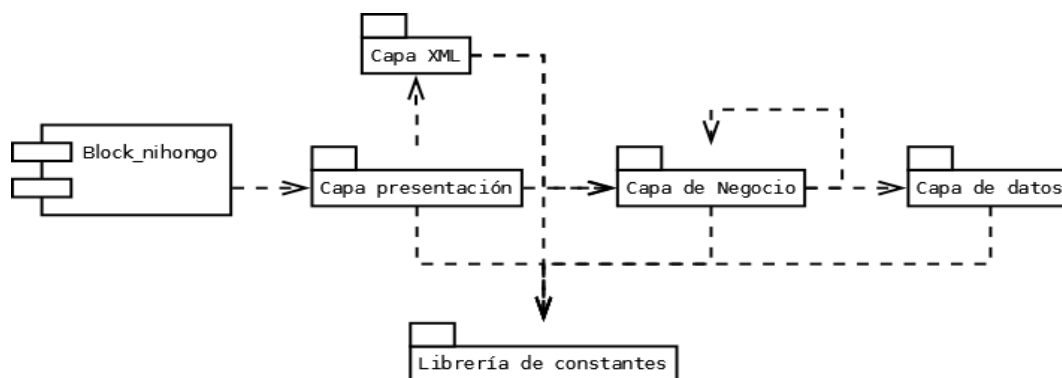
## 4.2.2 block\_nihongo

Este bloque da acceso a las diferentes actividades de aprendizaje de kanji ofrecidas por el módulo de actividades desarrollado. Contiene la mayor parte de código de las capas y toda la estructura de las tablas de la base de datos de todos los módulos. También se encarga de agregar a base de datos los diccionarios de kanji y vocabulario. El bloque tiene las siguientes características:

- Solamente puede ser agregado/mostrado en la página principal del curso.
- Muestra un enlace al listado de kanji y a la pizarra de entrenamiento de números.
- Tiene un configurador del idioma en el que vendrán traducidos las palabras y significado de los kanji japoneses de los diccionarios.
- El control de servidor *Kanji del Momento* junto con un configurador tanto global como instancial del control.

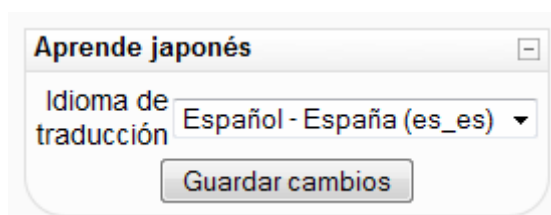
Cuando se instala el bloque en la plataforma *Moodle*, éste agrega a la base de datos todas las tablas relacionadas con los módulos de aprendizaje de japonés, además de leer los diccionarios de datos y agregarlos a las tablas. Para ello, debe modificar internamente algunas variables de la configuración de *PHP* ya que éstos, por defecto, bloquean su proceso al cabo de 60 segundos. Los campos son *max\_execution\_time*, *max\_input\_time*, *default\_socket\_timeout* y *mysql.connect\_timeout*. Se les cambia su valor a 50000 segundos para evitarnos cualquier problema de bloqueo. Para la gestión de los datos *XML*, se agrega una nueva capa de lógica a la arquitectura multicapa tal y como se

muestra en la figura 15. Esta capa se compone de dos clases, una que maneja el diccionario de kanji (*kanjixml*) y otra que manejará el diccionario de vocabulario (*vocabularyxml*). Tiene acceso a la capa de negocio, y solamente la capa de presentación tendrá acceso a esta capa.



**Figura 15 – Arquitectura de *block\_nihongo***

Una vez integrado el bloque en la página principal del curso, éste se encarga de comprobar si el usuario tiene configurado el idioma de traducción de las palabras japonesas. El proceso se hace internamente en el servidor. Si no lo encuentra, mostrará al bloque tal y como se muestra en la figura 16 y, si lo hace, tal y como se muestra en la figura 17 (por defecto). Aún así, desde la pantalla por defecto se podrá volver al configurador de idioma siempre y cuando se desee.



**Figura 16 – Configurador del idioma**



**Figura 17 – Principal con idioma seleccionado**

El configurador del idioma de traducción coge los idiomas instalados en *Moodle*, menos el idioma japonés. Una vez seleccionado el idioma, se realiza una llamada asíncrona al servidor que se encarga del proceso de almacenar la selección del usuario. La llamada al WS requiere los siguientes parámetros:

- **wsfunction**: Numérico entero. Valor 1 (Función de guardar idioma del usuario).
- **userid**: Numérico entero. Id del usuario conectado.
- **lang**: Cadena de caracteres. Idioma seleccionado en el formulario.

Si todo va correctamente, el navegador cliente se encarga de mostrar la pantalla principal del bloque inicializando *Palabra del Momento* con el idioma elegido y oculta el configurador.

El bloque puede ser configurado tanto instancialmente (como puede haber más de un curso que use el mismo tipo de bloque, cada bloque instalado es una instancia) como globalmente (afecta a todos los cursos que quieran instalar el bloque). Las opciones se muestran en la figura 18. La configuración global permite configurar los valores por defecto de las futuras instancias que se creen del bloque.

**Configurando un bloque Aprende japonés**

Activar Palabra del Momento ? : ☒

Activar autotransición: ☒

Tiempo de transición: 3000 milisegundos

Periodicidad: Cada día ▼

Modo: Por curso ▼

**Figura 18 – Configuración del bloque tanto instancial como global**

#### **4.2.2.1 Palabra del Momento**

La *Palabra del Momento* se encarga de mostrar al alumno un kanji y una palabra del vocabulario aleatorio durante un cierto tiempo con la finalidad de que el alumno descubra nuevo vocabulario o kanji y tenga interés por aprenderlo.

Es un control de servidor cuya lógica está diseñada con *jQuery* y el *plug-in* de generación de carruseles *jCarousel*. Mediante una conexión asíncrona, el control envía una petición al WS para obtener los elementos del control, dependiendo de la configuración del bloque. Los parámetros de entrada son los siguientes:

- **wsfunction**: Numérico entero. Valor 2 (Función de obtener *Palabra del Momento*).
- **userid**: Numérico entero. Id del usuario conectado.
- **courseid**: Numérico entero. Id del curso actual según *Moodle*.
- **groupid**: Numérico entero. Id del grupo en el que esté conectado el usuario según *Moodle*. Si la configuración del bloque indica que la modalidad del control es por curso en vez de por grupos, el valor es -1.
- **lang**: Cadena de caracteres. Idioma de traducción del vocabulario japonés.
- **periodicity**: Numérico entero. Código de la periodicidad de los caracteres según la configuración instancial del bloque.

El objeto *JSON* devuelto por el WS tras obtener los datos tiene la siguiente estructura:

```
{[
  {title: String, url: String, word: String, textsize: String, alternative:
String, description: String}
(, {}...)
]}
```

El significado de los campos son:

- **title**: Cadena de caracteres. Título que se muestra de cabecera del elemento. Está traducido al idioma que esté seleccionado *Moodle*.
- **url**: Cadena de caracteres. URL de destino si se hace *click* en el contenido del elemento. Por ejemplo, cuando se muestra el *Kanji del Momento*, el enlace te lleva a la tarjeta del kanji,
- **word**: Cadena de caracteres. Palabra importante, como una palabra japonesa o un kanji
- **textsize**: Cadena de caracteres. Tamaño de la palabra importante en pixels.
- **alternative**: Cadena de caracteres. Mensaje alternativo para la palabra importante. Al pasar el cursor del ratón por encima de la palabra se muestra el mensaje alternativo.
- **description**: Cadena de caracteres. Traducción de la palabra importante, traducido al idioma seleccionado por el usuario.

El objeto *JSON* es utilizado para montar las etiquetas HTML correspondientes al contenido de cada elemento del control. Hay dos elementos, *Kanji del Momento* y *Vocabulario del Momento*. Durante el tiempo de transición asignado en el configurador, el control automáticamente va rotando de un elemento a otro. Este tiempo de transición puede ser desactivado en el configurador, y se puede navegar por los distintos elementos mediante el uso de los controles de desplazamiento disponibles. Si durante el proceso automático se pulsa alguno de los botones de desplazamiento, queda desactivado el automatismo.

Mediante el configurador del bloque (ver figura 18) se puede adaptar el control de *Palabra del Momento* a nuestro gusto. Si el modo escogido es por curso, todos los usuarios ven el mismo kanji y palabra, mientras que si es escogido por grupo, los alumnos de cada grupo ven un kanji y palabra distinto al mismo tiempo. Las palabras se generan de forma aleatoria dependiendo de la periodicidad estipulada, que puede ser cada hora, cada día, cada semana o cada mes.

Las palabras escogidas durante cada periodo se almacenan en dos tablas de la base de datos: *mdl\_nihongo\_kotm*, para almacenar la información de *Kanji del Momento*; y *mdl\_nihongo\_votm*, para almacenar la información del *Vocabulario del Momento*.

### 4.2.3 Bloque principal (*mod\_nihongo*)

Tras la correcta instalación del bloque *block\_nihongo*, el módulo de actividades *mod\_nihongo* puede utilizarse correctamente. *Mod\_nihongo* proporciona una serie de herramientas útiles para el aprendizaje de japonés. Es aconsejado ocultar el acceso al módulo a los alumnos, ya que cuentan para acceder con los enlaces subministrados por *block\_nihongo* (ver figura 19). La diferencia entre un bloque y un módulo de actividad es que un bloque es un accesorio de una página, mientras que un módulo de actividad es todo un conjunto de páginas con el propósito de ofrecer una actividad al usuario. Los módulos de actividades pueden contener bloques, pero para ello hay que configurar *Moodle* para que permita la inserción de los bloques, además de preparar a las páginas de *mod\_nihongo* interesadas en tener bloques.





**Figura 19 – Icono diseñado y enlace al módulo de japones (Oculto a los estudiantes)**

*Mod\_nihongo* proporciona las siguientes utilidades:

- Un listado de kanji con la posibilidad de agrupación por tipos.
- Una página de tarjetas de estudio de kanji.
- Una pizarra de práctica de números en kanji.

En los siguientes apartados se describirán estas utilidades.

#### 4.2.3.1 Tarjetas de estudio de kanji (*cardvw*)

Las tarjetas de estudio de kanji presentan la posibilidad de estudiar estos caracteres de una forma simple y efectiva, obligando al estudiante a recordar la información del kanji antes de acceder a ella.

Se tomaron los siguientes requisitos para la cara delantera:

- Tiene que aparecer el kanji en grande.
- Tienen que aparecer, como máximo, tres palabras que contengan el kanji solicitado.

Para la cara trasera se tomaron los siguientes requisitos:

- Tienen que aparecer las lecturas del kanji, tanto las japonesas como las chinas (ver anexo 1 para más información sobre las lecturas).
- Tiene que aparecer el significado del kanji en el idioma escogido por el usuario.
- Tiene que aparecer información básica del carácter, tanto su grado JLPT (*Japanese Language Proficiency Test* – にほんごのうりよくしけん 日本語能力試験 [*Nihongo Nôryoku Shiken*]) como el número de trazos que tiene.
- Tiene que mostrarse la lectura en *hiragana* y la traducción de las palabras mostradas en la cara delantera, en el mismo orden de presentación.

La página que gestiona la creación de las tarjetas solicita dos parámetros en el URL (*Uniform Resource Locator*) de la página:

- ***id***: Número entero. Id del usuario conectado, para validar que el usuario conectado es el mismo que tiene guardado en sesión el sistema, además de si existe.
- ***kanji***: Número entero. Id del kanji según la tabla de la base de datos *mdl\_nihongo\_kanji*.

Una vez comprobado la validez de los datos administrados, el *script* de servidor se encarga de montar el código HTML y *Javascript* de la tarjeta solicitada. Cada requisito se resolvió de la siguiente manera:

- **Cara frontal (ver figura 20):**
  - Se obtiene el kanji correspondiente a la id de la base de datos (accediendo a *mdl\_nihongo\_kanji*) y mediante el empleo de CSS (*Cascading Style Sheet*) se modific

su tamaño. Estos datos se almacenan para usarlos en la parte trasera de la tarjeta. Un ejemplo del código HTML generado es el siguiente:

```
<div class="valign">
  <h1>握</h1>
</div>
```

- Se realiza una búsqueda en la tabla correspondiente al vocabulario con texto kanji de la base de datos (*mdl\_nihongo\_vocabulary\_k\_ele*) acotada a tres filas como máximo. Una vez obtenidos los resultados, se monta el HTML y se guardan los resultados para utilizarlos en la cara de detrás. Un ejemplo del HTML generado:

```
<ul>
  <li>握る</li>
  <!-- Otros elementos del vocabulario <li>...</li> -->
</ul>
```



Figura 20 – Parte delantera de la tarjeta de kanji

- **Cara trasera (ver figura 21):**

- Se recuperan todas las lecturas del kanji (alojadas en las tablas *mdl\_nihongo\_kanji\_kun*, para las lecturas japonesas, y *mdl\_nihongo\_kanji\_on*, para las lecturas chinas) y se monta una lista HTML estilizada con CSS. Las lecturas japonesas se representan en *hiragana*, mientras que las lecturas chinas se representan en *katakana*. Un ejemplo del código HTML generado:

```
<div>
  <ul>
    <li>アク</li>
    <!-- Otras lecturas <li>...</li> -->
  </ul>
</div>
```

- Se obtienen todos los significados disponibles almacenados en la tabla *mdl\_nihongo\_kanji\_meaning*, en el idioma escogido por el usuario (almacenado en la tabla *mdl\_nihongo\_user\_info*). Si no se encuentra traducción alguna en el idioma escogido por el usuario, por defecto se devuelve la traducción en inglés del carácter. Un ejemplo del código HTML generado:

```
<h2>
```

**Agarrar, coger, asir, empuñar, apretar con la mano**

- Con los datos almacenados tras mostrar por pantalla el kanji en la cara frontal, se obtiene la información deseada y se monta el HTML. Además se crea una pequeña ayuda para aclarar el significado de JLPT. Esta ayuda es un botón con icono generado por *Moodle* de forma genérica, y genera una ventana nueva que lee de forma formateada por los estilos propios de *Moodle*. Un ejemplo del HTML generado:

```
<div>
  <ul>
    <!-- Nivel JLPT -->
    <li>
      Nivel JLPT
      <!-- Código Botón ayuda Moodle -->
      : 1
    </li>
    <!-- Número de trazos -->
    <li>
      Número de trazos: 12
    </li>
  </ul>
</div>
```

- Se utilizan las id de las palabras del vocabulario obtenidas para la cara frontal. Se obtienen las lecturas de la tabla *mdl\_nihongo\_vocabulary\_r\_ele*, en unión a la tabla *mdl\_nihongo\_vocabulary\_re\_pri*, escogiendo la palabra con la lectura más utilizada. Para las traducciones se repite el mismo proceso que para obtener el significado del kanji, sólo que empleando para ello la tabla *mdl\_nihongo\_vocabulary\_gloss*. Una vez obtenida toda la información necesaria, se montan las listas en HTML y se estilan con CSS para mostrar los mensajes de forma organizada. Un ejemplo del código HTML generado:

```
<ul>
  <li>
    <span>にぎる</span>
    <span>To grasp, to seize, to mould sushi, to mold sushi</span>
  </li>
  <!-- Otros elementos del vocabulario <li>...</li> -->
</ul>
```

**Tarjetas de estudio de kanjis** ⓘ

|  |   |
|--|---|
| <div style="font-size: 2em; margin: 0;">           アク<br/>           にぎ.る         </div> |   |
| <b>Agarrar, coger, asir, empuñar,<br/>apretar con la mano</b>                            |   |
| Nivel JLPT ⓘ : 1   Número de trazos: 12  |   |
| おにぎり<br><br>にぎりめし<br>にぎりしめる  | Rice ball (often triangular,<br>sometimes with a filling and<br>wrapped in nori)<br>Rice ball<br>To grasp tightly |

[Volver al listado de kanjis](#)

### Figura 21 - Parte trasera de la tarjeta de kanji

Cada cara se genera dentro de bloques, ocultando por defecto el bloque de la parte trasera. Para crear el efecto de que realmente se dispone de una tarjeta con su cara frontal y trasera, se asignan eventos de ratón a la caja que envuelve las caras haciendo que cuando se realiza *click* en una cara, aparece la otra cara ocultando la mostrada hasta entonces.

#### 4.2.3.2 Listado de kanji (*kanjivw*)

El listado de kanji, como las tarjetas de estudio, es parte del módulo de actividades *mod\_nihongo*. Muestra por pantalla un numero de kanji organizados en forma de tabla de tal forma que el alumno pueda acceder a las tarjetas de estudio de los kanji deseados. Contiene bloques que mejoran la navegación y la utilidad del listado. Más adelante se comentarán estos bloques. Para que la página aceptara bloques, hubo que realizarse una serie de configuraciones ya que *Moodle*, por defecto, no permite que se muestren bloques en los módulos de actividades. Primero se tuvo que acceder a la pantalla del administrador de cursos y acceder a la opción *Ajustes de temas*, tal y como se muestra en la figura 22. Una vez seleccionado, buscar la opción marcada con el código *Moodle showblocksonmodpages*. Por defecto está desactivado, por lo que hay que marcarla (ver figura 23).

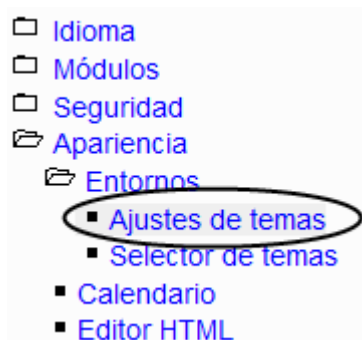


Figura 22 – *Ajustes de temas* en el menú del administrador

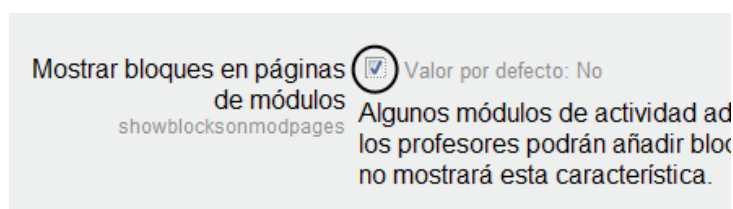


Figura 23 – Marcar la casilla *showblocksonmodpages*

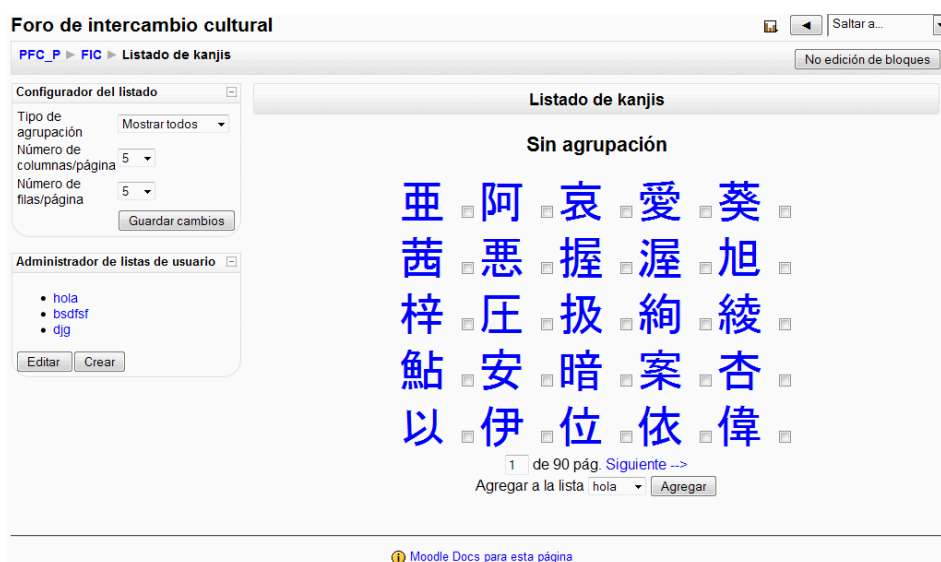
Una vez preparado *Moodle*, se tiene que preparar la página para que ésta inserte correctamente los bloques. Interesa configurar la barra del menú para que muestre un botón para la edición de bloques (ver figura 24, arriba a la derecha) y que los bloques por defecto sean el configurador del listado (*block\_kanji\_config*) y el administrador de listas de usuario (*block\_userlist\_nihongo*).

Desde el servidor se prepara la página para que, desde el navegador cliente, se genere el listado mediante el empleo de *AJAX*. Para ello, se genera una plantilla *Javascript*, que será capaz de traducir el *plug-in jTemplate* una vez obtenidos los datos del WS. Esta plantilla es adaptable a diferentes situaciones en las que se pueden encontrar los datos recibidos. La llamada WS requiere los siguientes parámetros:

- ***wsfunction***: Numérico entero. Valor 1 (Función de obtener listado de kanji).
- ***userid***: Numérico entero. Id del usuario conectado.
- ***groupby***: Numérico entero. Código de la agrupación de kanji seleccionada. 0 por defecto.

- **numcols**: Numérico entero. Número de columnas por página. 6 por defecto.
- **numrows**: Numérico entero. Número de filas por página. 3 por defecto.
- **selected**: Numérico entero. Código de la sub-agrupación de kanji seleccionada. 1 por defecto.
- **numpage**: Numérico entero. Número de página de la cual recuperar los datos.

Una vez realizada la petición se validan los datos y se devuelve un objeto *JSON* que es tratado por la plantilla a fin de construir un listado como el mostrado en la figura 24 o en la figura 25.



**Figura 24 – Listado de kanji sin agrupación**



**Figura 25 – Listado de kanji agrupado por niveles de JLPT**

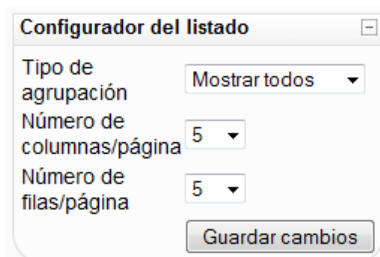
Como se puede comprobar en las figuras, aparecen varios controles dentro del listado que facilitan la navegación por el listado. Debajo del listado encontramos un control de páginas en el que poder desplazarnos tanto página por página (mediante los enlaces de Siguiente y Atrás) como ir directamente a la página deseada utilizando el campo de texto alojado entre estos enlaces. Encima del listado, en los casos que la agrupación seleccionada tenga subgrupos, aparecerá un desplegable indicando los distintos sub-agrupaciones disponibles. Una vez escogido, se mostrarán los caracteres del subgrupo (la figura 25 es un ejemplo de sub-agrupación). Cada carácter mostrado por pantalla

tiene asignado un enlace a la tarjeta de estudio correspondiente, pudiéndose profundizar el estudio de cada carácter interesado. También, si el usuario dispone de listas personales de kanji (ver sección *block\_userlist\_nihongo* más adelante para más información), aparecen unas casillas de chequeo al lado de cada carácter, útiles para agregar nuevos caracteres a la lista deseada por el usuario. Esta lista de usuario se podrá mostrar cuando lo desee el usuario por pantalla, y entonces la acción de la selección de casillas corresponderá a la eliminación de la lista de los caracteres deseados. Todos estos procesamientos se hacen de forma asíncrona mediante el uso de *AJAX*:

- Para agregar caracteres en las listas predefinidas, realiza una llamada al WS con los siguientes parámetros:
  - ***wsfunction***: Numérico entero. Valor 3 (Función de añadir kanji a una lista de usuario).
  - ***userid***: Numérico entero. Id del usuario conectado.
  - ***kanjilist***: Cadena de caracteres. Listado de las id de los kanji a agregar separados por comas (,).
  - ***userlistid***: Numérico entero. Id de la lista del usuario.
- Para eliminar caracteres de una lista predefinida, realiza una llamada al WS con los siguientes parámetros:
  - ***wsfunction***: Numérico entero. Valor 4 (Función de eliminar kanji de una lista de usuario).
  - ***userid***: Numérico entero. Id del usuario conectado.
  - ***kanjilist***: Cadena de caracteres. Listado de las id de los kanji a eliminar separados por comas (,).
  - ***userlistid***: Numérico entero. Id de la lista del usuario.

### Configurador del listado de kanji (*block\_kanji\_config*)

Este bloque se encarga de la configuración del listado de kanji, dando la posibilidad a cada usuario de tener una página inicial a su gusto, sin que eso afecte al resto de usuarios. Dentro de las posibilidades que ofrece configurar está el tipo de agrupación de kanji, el número de columnas por página o el número de filas por página (ver figura 26). Esta información será almacenada en la tabla *mdl\_nihongo\_user\_info*.



**Figura 26 – Página principal**

Las opciones disponibles en los desplegables son asignadas de forma rígida, por lo que para agregar nuevas agrupaciones es necesario modificar código. El número seleccionable tanto de columnas como de filas es un rango compuesto deliberadamente entre 1 y 10, para evitar problemas de integridad de la página. Por defecto, el tipo de agrupación es *mostrar todos*, el número de columnas por página es 6 y el número de filas por página es 3.

Para almacenar los datos se realiza una conexión asíncrona mediante *AJAX*. La llamada para el WS

requiere los siguientes parámetros:

- **wsfunction**: Numérico entero. Valor 1 (Función de obtener listado de kanji).
- **userid**: Numérico entero. Id del usuario conectado.
- **groupby**: Numérico entero. Id de la agrupación del listado seleccionado.
- **numcols**: Numérico entero. Número de columnas por página seleccionada.
- **numrows**: Numérico entero. Número de filas por página seleccionada.

Una vez procesado con éxito estos cambios, se recurre a la recarga del listado de kanji con las configuraciones deseadas.

### Administrador de listas de usuario (*block\_userlist\_nihongo*)

Este bloque se encarga de administrar listados predefinidos de kanji únicos por usuario. Se permite crear listas, renombrar listas y eliminar listas. La gestión del contenido de kanji se realiza desde el propio listado de kanji, como se describió anteriormente. El bloque hace un uso muy extenso de la conexión asíncrona, haciéndose ver en la creación, edición y eliminación de las listas. Las listas se almacenan en la tabla *mdl\_nihongo\_user\_list*, mientras que los kanji de cada lista se almacenan en la tabla *mdl\_nihongo\_user\_kanji\_list*. Se programó para estar preparado por si se deseaba agregar el bloque en otros listados, mediante un tipo de lista almacenado en la tabla *mdl\_nihongo\_user\_list*.



**Figura 27 – Página principal con listas**

La figura 27 muestra la página principal del bloque ya con listas creadas. Cada una contiene un enlace que realiza una petición asíncrona para mostrar en la pantalla del listado los caracteres asignados a la lista. La petición del WS requiere los siguientes parámetros:

- **wsfunction**: Numérico entero. Valor 1 (Función de obtener listado de kanji).
- **userid**: Numérico entero. Id del usuario conectado.
- **groupby**: Numérico entero. Id de la agrupación seleccionada. En este caso, id para el tipo lista predefinida (3).
- **numcols**: Numérico entero. Número de columnas / página seleccionada. Por defecto 6.
- **numrows**: Numérico entero. Número de filas / página seleccionada. Por defecto 3.
- **selected**: Numérico entero. Id de la lista predefinida.
- **numpage**: Numérico entero. Número de la página solicitada.
- **title**: Cadena de caracteres. Nombre de la lista predefinida.

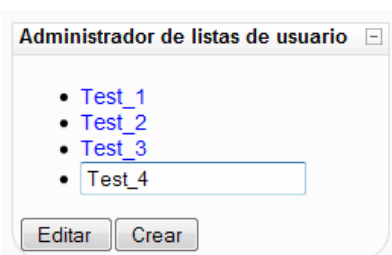
Esta petición devuelve un objeto *JSON* comprensible para la plantilla del listado de kanji, obteniendo una pantalla como la ofrecida en la figura 28. Además, el listado cargado ofrece un botón para eliminar de la lista los kanji seleccionados mediante las casillas, como se explicó

anteriormente.



**Figura 28 – Listado de una lista predefinida de kanji**

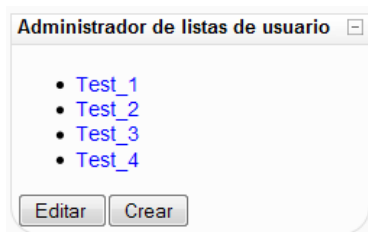
Este bloque, además, administra las listas de usuarios. En primera instancia, nos ofrece además del listado de listas (si éstas existen), botones que se encargan de dos tareas: crear una nueva lista o editar las listas ya existentes. Si no hay listas que editar, el botón de edición queda oculto a la espera de la creación de una nueva lista por parte del usuario. Al pulsar el botón de crear, dinámicamente se genera un nuevo elemento de lista con un campo de texto a rellenar, tal y como se muestra en la figura 29. Una vez rellenado el campo solamente hay que hacer *click* en cualquier otro elemento de la ventana y, mediante una llamada al WS mediante *AJAX*, se realiza el proceso de inserción de la lista con el nombre administrado. Tanto el navegador cliente como el servidor se encarga de comprobar que no se repita el nombre de la lista del usuario, dando un mensaje de error en caso de que ya exista dicha lista. Si se produce el error, la lista desaparece. Los parámetros que requiere la llamada al WS son los siguientes:



**Figura 29 – Formulario para agregar una nueva lista**

- **wsfunction**: Numérico entero. Valor 1 (Función de crear nueva lista de usuario).
- **userid**: Numérico entero. Id del usuario conectado.
- **userlist**: Cadena de caracteres. Nombre del listado de kanji definido.
- **typelist**: Cadena de caracteres. Tipo de listado. Para los listados de kanji, el valor es 'k'.

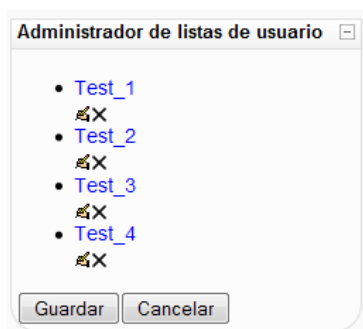
Una vez completado el proceso correctamente, el campo de texto será reemplazado por un enlace, tal y como se muestra en la figura 30.



**Figura 30 – Lista agregada con éxito**



El botón de editar modifica el listado dinámicamente agregando botones a cada lista que representan la edición y eliminación de cada una de las listas (ver figura 31). Se realiza una copia de seguridad del listado de kanji por si se decide cancelar la edición y recuperar los datos. Los botones de editar y crear se suplantán por botones de guardar y cancelar, el primero de ellos encargado de procesar todos los cambios generados por pantalla y el segundo, de deshacer todos los cambios realizados. El icono de una mano es el indicativo de renombrar la lista, mientras que la cruz es el indicativo de eliminar la lista. El proceso previo de edición y eliminación se produce dinámicamente mediante el uso de *Javascript* y la librería *jQuery*. Todos los cambios que se realicen se almacenan en memoria hasta que no se pulsa el botón guardar o cancelar. Cuando se pulsa el icono de edición, el nombre de la lista es substituido por una caja de texto con el nombre de la lista en su interior. Si ésta es modificada, se vuelve a mostrar el enlace de la lista y, internamente en el código HTML, se le asigna a la lista el atributo *renamed*. Si se pulsa el icono de eliminación la lista se vuelve oculta y se le asigna el atributo *deleted*.



**Figura 31 – Edición de las listas de usuario.**

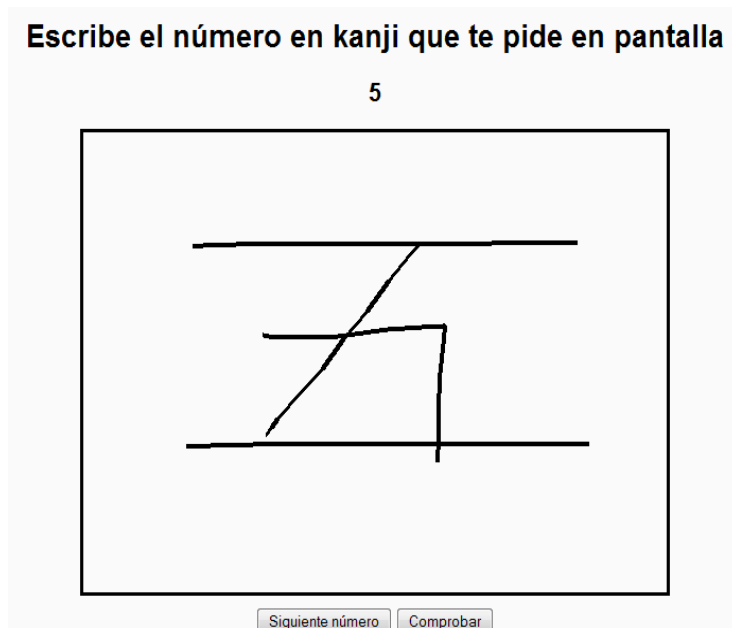
Una vez realizados los cambios, si se pulsa el botón de cancelar, la lista copiada por seguridad suplanta a la editada, eliminando el rastro de copia. Para este proceso no es necesario el uso de los WS, se realiza dinámicamente mediante *Javascript*. Si se pulsa el botón de guardar, entonces se recorre la lista modificada en busca de las listas que tengan asignados los atributos *renamed* y *deleted*. Dependiendo del tipo de edición aplicada a la lista, se realizará una llamada asíncrona al WS, por cada lista editada. Las llamadas al WS, dependiendo del tipo de edición, utilizan los siguientes parámetros:

- ***wsfunction***: Numérico entero. Valor 2 (Función de eliminación de lista de usuario) o valor 3 (función de renombramiento de lista de usuario).
- ***userid***: Numérico entero. Id del usuario conectado.
- ***idlist***: Numérico entero. Id de la lista predefinida de kanji a modificar.
- Si se renombra:
  - ***userlist***: Cadena de caracteres. Nuevo nombre de la lista.
  - ***typelist***: Cadena de caracteres. Tipo de listado. Para los listados de kanji, el valor es 'k'.
- Si se elimina:
  - ***typelist***: Cadena de caracteres. Tipo de listado. Para los listados de kanji, el valor es 'k'.

Una vez terminado todo el proceso, se recupera la pantalla tal y como se muestra en la figura 30, pero con las listas editadas y almacenados los cambios en memoria.

#### 4.2.3.3 Pizarra de kanji (trainvw)

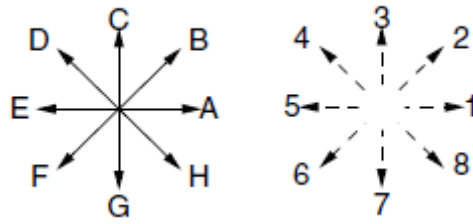
La última de las herramientas a explicar del módulo de actividades corresponde a una pantalla que realiza pruebas de conocimiento de escritura de números del 1 al 10 en kanji mediante el uso de una pizarra realizada en *Flash*. La página genera un número aleatorio y solicita al usuario que dibuje el carácter correspondiente (ver figura 32). Una vez dibujado, pulsando al botón de comprobar, se comprueba que el carácter dibujado es el correspondiente al solicitado. Si lo reconoce, se muestra en pantalla un mensaje de éxito y vuelve a solicitar otro número distinto. Si no lo reconoce, avisa del error y solicita la introducción de nuevo del carácter. El bucle es infinito, por lo que se pueden realizar tantas pruebas como se desee. Si se desconoce el carácter, pulsando el botón de *Siguiente número* se obtiene otro número con el que probar.



**Figura 32 – Pizarra de kanji con el kanji que representa el número 5 dibujado**

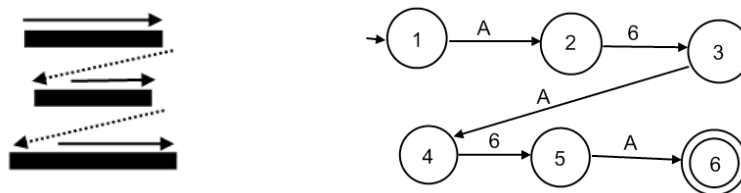
Para el reconocimiento de caracteres, se realizó un estudio de diferentes fuentes sobre la metodología a seguir. Basándome en distintos estudios, el más interesante propuesto por Junko Tokuno, Nobuhito Inami y otros compañeros [29], aconsejaban implementar un reconocedor de caracteres japoneses por sus trazos y su ordenación (ver anexo 1 para ampliar información sobre el orden de trazos). Estos estudios explicaban la idea de implementar cadenas ocultas de Markov (HMM – *Hidden Markov Models*) por cada tipo de trazo, subtrazo o salto de trazo existente en cada carácter. Las cadenas ocultas de Markov son un modelo estadístico cuya función es la de *determinar los parámetros desconocidos* (de ahí el nombre de ocultos) *de dicha cadena a partir de los parámetros observables* [30].

Se tuvo que descartar esta implementación porque se requerían unas estadísticas iniciales obtenidas a partir de miles de pruebas que no se disponían. La alternativa fue hacer uso de máquinas de estado finitas. Las máquinas de estado finitas o autómatas finitos son *un modelo matemático de un sistema que recibe una cadena constituida por símbolos de un alfabeto y determina si una cadena pertenece al lenguaje que el autómata reconoce* [31]. Los símbolos del alfabeto están formados por dos grupos de símbolos, unos formados por la interpretación de la dirección del trazo y otros formados por la interpretación de la dirección de dejar de dibujar un trazo, desplazarse a otro lugar para continuar el dibujo del kanji y comenzar a dibujar el siguiente trazo. En total 16 símbolos. La lista de símbolos se pueden contemplar en la figura 33.



**Figura 33 – A la derecha, símbolos relacionados con la dirección del dibujo del trazo; a la izquierda, símbolos relacionados con la dirección de los saltos de trazo**

Un ejemplo práctico puede mostrarse con la figura 34 con el carácter 三 (tres). Se respeta el tipo de flecha representada en la figura 33. Este carácter, siguiendo el orden de trazos reglamentario, se representaría en forma de cadena de símbolos como “A6A6A”. Otro carácter, como □ (boca, entrada) se representaría como “G3AG5A”. Estas representaciones idílicas se usan como elementos de un diccionario de datos de los caracteres a reconocer.

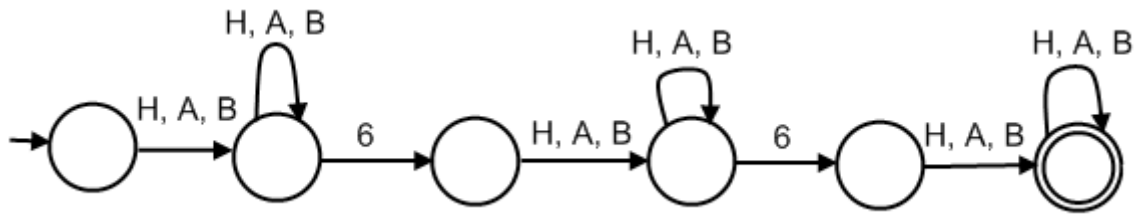


**Figura 34 – A la izquierda, carácter 3 según el orden de trazos obligatorio, desde arriba hacia abajo y de izquierda a derecha. A la derecha, máquina de estados finita ideal del carácter 3.**

Estas reglas, a nivel ideal, son fácilmente desarrollables, pero la escritura a mano alzada proporciona los siguientes problemas:

- **El trazado no es uniforme.** Mientras uno dibuja, es difícil mantener la dirección de un trazo en todos los puntos del trazado. Es fácil cometer pequeñas desviaciones durante el trazado. Este problema es extensible al hecho de que un trazo puede estar compuesto con más de una dirección, como se ha comentado anteriormente con el carácter □. Al poder haber mal-interpretaciones de direccionamiento, el tratamiento de este tipo de trazos se hace más complejo de lo que lo ideal presenta.
- **El tamaño del carácter es variable según la persona.** En todo el tamaño de la pizarra, el carácter puede ser dibujado del tamaño que sea capaz el usuario, pudiendo ocasionar problemas a la hora de interpretar la dirección de los saltos de trazo.
- **Estilo de trazado del usuario.** Existen caracteres con trazos que algunas personas dibujan en una dirección u otra, aceptadas por las reglas de escritura. Por ejemplo, el primer trazo del carácter 穴 (agujero) puede ser interpretado tanto como “G” como “H”, entre otras posibles variantes igualmente aceptadas de los consiguientes trazos. Esto hace tener que preparar al diccionario de datos con la posibilidad de múltiples representaciones de un mismo carácter, dificultando la tarea de creación de la máquina de estados deseada.

Por ello se plantea la generación de máquinas de estado cuyos nodos contengan no solamente información del proceso ideal de la creación de un carácter sino, además, enlaces extra que contengan información de los símbolos más cercanos como posibles estados, menos de los saltos que siguen siendo obligatorios los solicitados idealmente. También se tiene en cuenta el hecho que, al dibujarse un carácter a mano alzada, no se obtiene un sólo símbolo por trazo dibujado sino toda una serie de símbolos calculados cada *frame* que pasa, por lo que hay que generar bucles para estacionar a un trazo en un mismo estado hasta un cambio dirección importante o salto de trazo. Un ejemplo de la máquina de estado planteada sería la que se muestra en la figura 35.

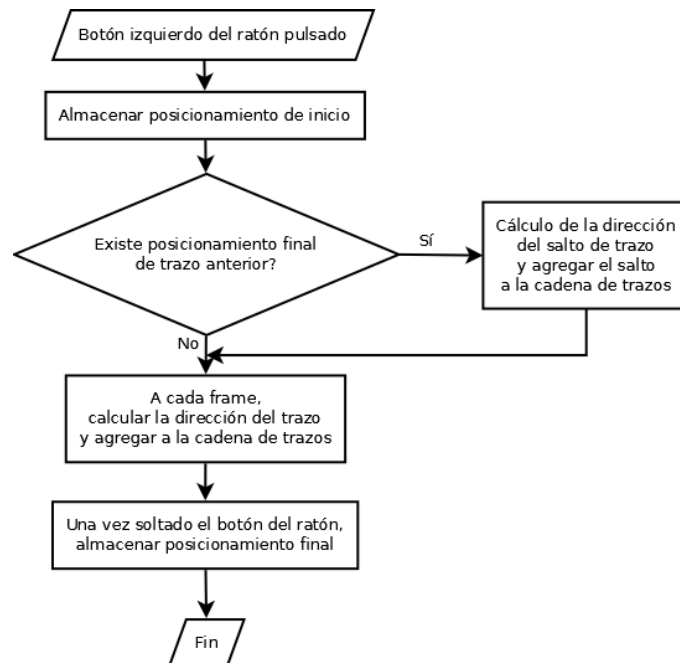


**Figura 35 – Máquina de estados implementada para reconocer el carácter 三**

A continuación describimos el funcionamiento de la pizarra Flash y su propósito en este detector de caracteres japoneses.

### La pizarra Flash

La pizarra hecha en *Flash* se encarga de generar una cadena de direcciones de trazos y saltos de trazos necesaria para el futuro reconocimiento del carácter. Para ello, se pueden distinguir dos funcionalidades. En la figura 36 se puede comprobar el grafo del algoritmo utilizado para la construcción de la cadena de símbolos. Para el cálculo de cada símbolo, se usa el punto de inicio  $P_1 = (x_1, y_1)$ , punto donde el ratón está posicionado al comienzo de un *frame*; y el punto  $P_2 = (x_2, y_2)$ , punto donde el ratón está posicionado al finalizar el *frame*. El punto  $P_1$  se obtiene a partir del punto  $P_2$  del *frame* anterior.



**Figura 36 – Diagrama de flujo del proceso de la pizarra para montar la cadena de trazos**

Una vez obtenidos los puntos, se obtiene el vector relacionado mediante la formula siguiente:

$$V(v_x, v_y) = (x_2 - x_1, y_2 - y_1)$$

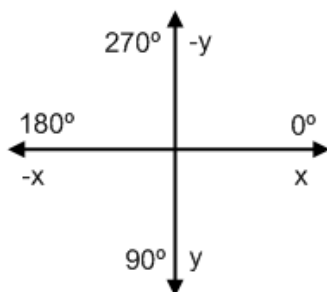
Una vez obtenido el vector  $V$ , se normaliza el vector para un cálculo de la dirección más fácil:

$$V_n = \left( \frac{v_x}{\sqrt{v_x^2 + v_y^2}}, \frac{v_y}{\sqrt{v_x^2 + v_y^2}} \right)$$

Cuando el vector dirección está normalizado, calculamos el ángulo del vector en radianes para más adelante obtener el símbolo deseado. Para ello hay que tener en cuenta cómo *Flash* interpreta el eje de las coordenadas cartesianas. El eje vertical es interpretado al revés, tal y como se muestra en la

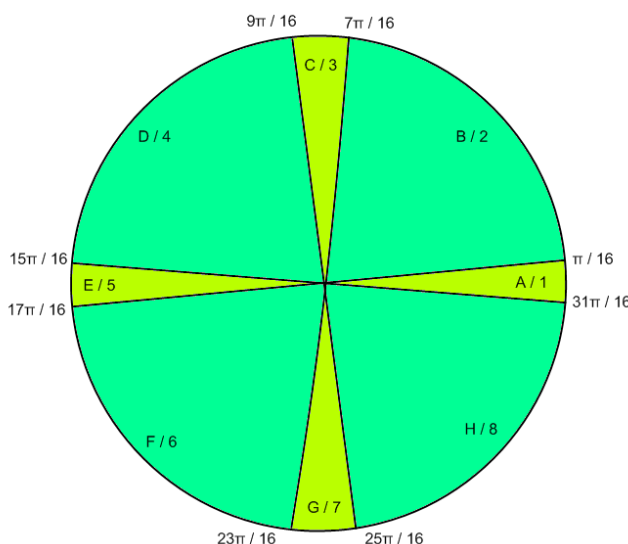
figura 37. *Flash* a los ángulos superiores a 180° los convierte a su representación negativa, por lo que hay que tratar estos valores y obtener su representación positiva. Para obtener el ángulo en radianes deseado en base a las coordenadas cartesianas estándar, se utiliza la siguiente fórmula:

$$ang^{\circ} = \arctan\left(\frac{-v_y}{v_x}\right)$$



**Figura 37 – Eje de coordenadas cartesianas interpretadas por Flash**

Una vez tenemos el ángulo de la dirección, comprobamos en nuestro rango de valores a qué símbolo corresponde. Tras distintas pruebas, la siguiente figura muestra el rango de valores marcado para cada símbolo del diccionario:



**Figura 38 – Disposición del rango de ángulos por símbolo del diccionario**

Esta disposición de rangos facilita la interpretación de la dirección de los saltos, principalmente, ya que una repartición equitativa de rangos ocasionan errores positivos que afectan a la interpretación del carácter.

Una vez localizado el rango que corresponde al ángulo obtenido, se obtiene el símbolo deseado para la ocasión (tanto si es una dirección de trazo como de salto) y se agrega al final de una cadena preparada a almacenar toda una lista de símbolos que después el interprete de máquinas de estados utilizará para detectar qué carácter es el dibujado. Eso implica que tendremos, por cada *frame*, un símbolo del trazo mientras se dibuje, pero un sólo símbolo por salto de trazo realizado. Un ejemplo de caso se puede comprobar en la figura 39.



**Figura 39 – A la izquierda, un kanji dibujado en la pizarra. En el centro, los diferentes trazos que se van dibujando *frame a frame* y su dirección, además de los saltos, y como interpreta cada subtrazo. A la derecha, la cadena de símbolos resultante**

La pizarra *Flash* se incrusta dentro del HTML de la página como un objeto, pero mediante *Javascript* no se puede acceder directamente a ningún dato que pueda almacenar el objeto *Flash*. Para poder interactuar *Javascript* y *Flash*, éste debe implementar funciones que acaben incrustándose a una llamada *Javascript* mediante el empleo de la función *ExternalInterface.addCallBack()*. Este método requiere tres parámetros específicos:

- **methodName:** Cadena de caracteres. El nombre del método que tiene la función en *Javascript*.
- **instance:** Objeto. Objeto que tiene el valor *this* en *Flash*. Puede ser nulo.
- **method:** Función. Función *ActionScript* que realiza el proceso de la llamada *Javascript*.

Para la pizarra se crearon dos llamadas externas, una para limpiar la pizarra y reiniciar los datos (*CleanWhiteboard*) y otra para pasar la cadena de símbolos a una variable JavaScript (*GetStrokeChain*).

A continuación se explica el proceso realizado por el navegador a nivel de *Javascript*

### La pantalla de la pizarra

La pizarra *Flash* se incrusta en el HTML con el código siguiente, generado automáticamente por *Adobe Flash*:

```
<OBJECT classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
  codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflas
h.cab#version=10,0,0,0"
  WIDTH="550" HEIGHT="400" id="pizarra" ALIGN="middle">
  <PARAM NAME=movie VALUE="flash/pizarra.swf">
  <PARAM NAME=quality VALUE=low>
  <PARAM NAME=bgcolor VALUE=#FFFFFF>
  <PARAM NAME=wmode VALUE=transparent>
  <EMBED src="flash/pizarra.swf"
    quality=low
    swliveconnect="true"
    bgcolor=#FFFFFF WIDTH="550" HEIGHT="400"
    name="pizarra"
    ALIGN="middle"
    TYPE="application/x-shockwave-flash"
    PLUGINSOURCE="http://www.macromedia.com/go/getflashplayer"
    wmode="transparent">
  </EMBED>
</OBJECT>
```

Cosas a tener en cuenta de este etiquetado:

- La etiqueta OBJECT es una etiqueta estándar con XHTML, y es usado por navegadores parecidos a *Internet Explorer* para poder cargar *Flash*, mientras que la etiqueta EMBED no es estándar con XHTML y sólo es usada por navegadores similares a *Netscape*. La utilización de ambas etiquetas nos asegura una gran compatibilidad de navegadores en los que pueda ejecutarse *Flash*.
- El parámetro *swliveconnect* (etiqueta EMBED) con valor *true* permite a *Javascript* ejecutar funcionalidades de *Flash*, como las funciones con llamada externa comentadas en el apartado anterior, mediante el empleo de la id del objeto como nombre de acceso al objeto *Flash*, en este caso *pizarra*. Un ejemplo de llamada *Javascript* al objeto *Flash* es:

```
document.pizarra.CleanWhiteboard();
```

Cuando se pulsa el botón de *Validar*, se obtiene de la pizarra *Flash* la cadena de símbolos mediante la llamada al objeto *Flash* *document.pizarra.GetStrokeChain()* y se produce una llamada asíncrona al WS con los siguientes parámetros de entrada:

- ***wsfunction***: Numérico entero. Valor 2 (Función de comprobar validez del kanji).
- ***userid***: Numérico entero. Id del usuario conectado.
- ***number***: Numérico entero. Número mostrado por pantalla para comprobar validez.
- ***strokelist***: Cadena de caracteres: Cadena de símbolos obtenida de la pizarra *Flash*.

Una vez procesada la llamada ésta devolverá un objeto JSON con la estructura siguiente:

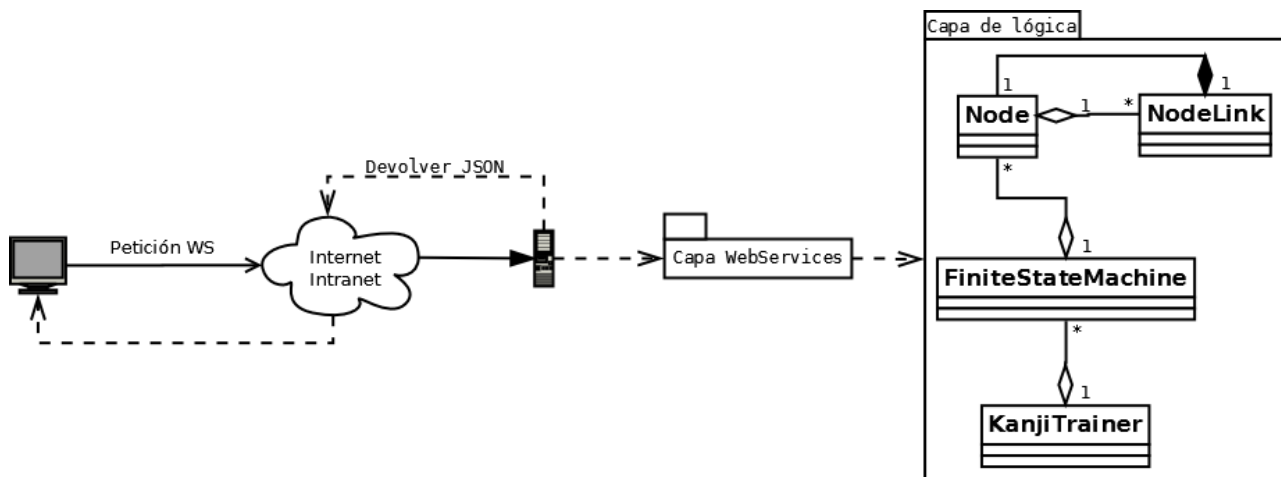
```
{kanji: Number}
```

Si *kanji* tiene valor 1, es que el kanji es correcto y cambia el número de comprobación. Si *kanji* tiene valor 0, el kanji es incorrecto, limpia la pantalla y te vuelve a pedir el mismo número, tantas veces como sea necesario.

A continuación se describe el proceso del WS y cómo está montada la arquitectura de capas para este caso especial.

### Arquitectura del script de servidor

La arquitectura en capas para el entrenamiento de kanji modifica la estructura estándar utilizada, ya que ésta no requiere el uso del acceso a base de datos, y la capa de controlador queda suplantada por una capa de lógica especializada en la generación y manipulación de máquinas de estado finitas, como se puede comprobar en la figura 40.



**Figura 40 – Diagrama de la arquitectura de capas del entrenamiento de kanji. El proceso es parecido al diagrama de la figura X, sólo que ya no existe capa de datos.**

La capa de lógica está compuesta por cuatro clases, aunque solamente 1 de ellas es pública a la capa de WS. El objeto *Node* representa la funcionalidad de los nodos-estado de una máquina de estados finita. Puede contener un número infinito de objetos *NodeLink*, que representan el enlace a otros nodos-estado. Cada *NodeLink* es capaz de tener como destino otro objeto de tipo *Node* ya generado. Los objetos *Node* son almacenados en objetos de tipo *FiniteStateMachine*, clase que representa la funcionalidad de una máquina de estados finita. Objetos de esta clase son almacenados en la clase *KanjiTrainer*, encargada de realizar el proceso de entrenamiento de kanji, preparando las máquinas de estado de los caracteres disponibles, y realizando el proceso de reconocimiento de caracteres empleando las máquinas de estados de caracteres ya generados previamente.

El WS se encarga de proporcionarle al objeto *KanjiTrainer* una serie de kanji con su disposición de trazos ideal, y éste se encarga de realizar máquinas de estado de cada kanji, agrupándolos por número de trazos. En el caso de los 10 primeros números japoneses se requerirán tan sólo 5 máquinas de estado, compuestos por caracteres de uno a cinco trazos. En el caso de unir en una máquina múltiples kanji distintos, incluidos aquellos duplicados con un trazado alternativo, se unen en un proceso que escoge el mejor camino de las máquinas a unir, reduciendo estados y enlaces y, con ello, ahorrar espacio de memoria. Es un proceso poco costoso para esta cantidad de caracteres que se repite en cada petición a esta función.

Una vez entrenada la máquina de estados, se le proporciona al objeto *KanjiTrainer* el listado de caracteres obtenido a través de la pizarra y, mediante el recuento trazos calculado, escoge la máquina de estados adecuada y lo comienza a recorrer símbolo a símbolo, haciendo el recorrido propuesto siempre que pueda navegar al nodo siguiente. Si en algún momento del recorrido no encuentra destino con el símbolo en el que se encuentra, devolverá 0 como resultado de la búsqueda, indicando con ello que hay un error. Si llega al final de la cadena de símbolos y no se encuentra en un estado final, también devuelve 0. Si se encuentra en un estado final, devuelve el carácter que tenga asociado a ese estado final. Con ese carácter, se comprueba si el que solicita la página es el que encuentra y generará un objeto JSON como el comentado en el apartado anterior.



## 5. Resultados y conclusiones

En este capítulo se exponen las pruebas realizadas, las conclusiones y las mejoras en el futuro.

### 5.1 Pruebas realizadas y resultados

En este apartado se presentan una serie de pruebas tanto de compatibilidad como test de velocidad. Para la realización de las pruebas de compatibilidad, se ha hecho uso del navegador *Mozilla Firefox* 3.5.3 y de *Windows Internet Explorer* 8.0. Todas las imágenes mostradas a lo largo de la memoria se han realizado utilizando *Firefox*. Las imágenes siguientes corresponden a la versión de *Internet Explorer*:



Figura 41 – Distintas representaciones de los bloques creados vistos por *Internet Explorer*

A continuación se muestra una tabla de test de velocidad de los procesos asíncronos producidos por los bloques y módulos:

| <b><i>Bloque/Módulo</i></b> | <b><i>Llamada</i></b>                             | <b><i>Tiempo promedio</i></b> |
|-----------------------------|---|-------------------------------|
| block_nihongo               | Guardar idioma traducción                         | 488 ms                        |
| block_nihongo               | Carga Palabra del Momento                         | 717,93 ms                     |
| mod_nihongo (kanjivw)       | Carga Listado de kanji                            | 417 ms                        |
| block_kanji_config          | Guardar configuración y recargar listado de kanji | 446,5 ms                      |
| block_userlist_nihongo      | Crear lista de usuario                            | 491 ms                        |
| block_userlist_nihongo      | Editar conjunto listas de usuario                 | 552 ms                        |
| mod_nihongo (trainvw)       | Comprobar validez de un carácter                  | 886,3 ms                      |

Se comprueba con ello que el tiempo de espera es tan poco que apenas es apreciable en la mayoría de los casos, menos de 1 segundo de proceso, haciendo que el uso de las nuevas utilidades no sea molesto por la espera de ver qué va a suceder.

## 5.2 Conclusiones y mejoras

1. La plataforma *Moodle* propone una serie de grandes herramientas a estudiantes y a profesores por igual. Pero la pobreza de la documentación de desarrollo hace que aprender a crear nuevos elementos para esta plataforma, un trabajo costoso que penaliza gran parte del tiempo de desarrollo del proyecto. Aún así se ha conseguido ofrecer al cliente una serie de herramientas bastante completas que ayudarán al estudiante a reforzar sus conocimientos de los caracteres japoneses.
2. *jQuery* ha demostrado ser un *framework* muy potente, ligero e intuitivo de manejar. *Moodle* ganaría mucho a nivel de desarrollo si implementara esta librería como estándar.
3. Una mejora interesante para la *Palabra del Momento* sería la creación de nuevas categorías, como *Refrán del Momento* o *Gramática del Momento*. Sería interesante para ello la creación de un formulario de categorías, pero cada subgrupo de la Palabra del Momento tendría que re-configurarse como componentes, de tal forma que se pudieran conectar y desconectar del control sin necesidad de modificar código manualmente.
4. Sería interesante agregar a la información de los kanji la información de los radicales. Los radicales se pueden considerar el elemento raíz del kanji. La principal causa de su descarte para el proyecto es la falta de la total existencia de los radicales como caracteres *Unicode*.
5. Ya que existe actualmente un listado de kanji, se podría crear un listado de vocabulario basándose en los datos disponibles en base de datos, ya preparada para cualquier situación. También se le podría agregar a este listado el bloque de listas predefinidas de usuario, que con pequeñas modificaciones de código estaría listo para aceptar listados de vocabulario.
6. Ahora mismo se aprovecha una cantidad limitada de kanji y vocabulario que ofrecen los diccionarios XML. Sería interesante almacenar en base de datos toda la información de todos los caracteres disponibles, aunque esto requiere mucho tiempo de procesamiento (puede llegar a durar horas).
7. La creación de formularios que controlen la información tanto de kanji como de vocabulario, como su significado, ayudaría a mejorar el servicio de los bloques creados, además de ayudar a los alumnos a estudiar con información actualizada.

8. Para el reconocimiento de caracteres sería necesario transportar la información de las máquinas de estado generadas por el entrenamiento de caracteres en algún archivo lógico, sea fichero físico o en base de datos, para no tener que repetir cada vez el proceso de entrenamiento de las máquinas de estado, aligerando el tiempo de proceso futuro de reconocimiento de caracteres. También sería interesante crear una tabla nueva para los kanji que contenga los diferentes órdenes de trazo idílicos de los que están compuestos, así podría ampliarse fácilmente la biblioteca de caracteres reconocidos por la máquina de estado sin tener que modificar código del módulo.
9. También sobre el reconocimiento de caracteres, una idea para mejorar el reconocimiento sería controlar el tamaño de los trazos que requiere cada carácter. Actualmente el proceso solamente se enfoca en el orden de trazos, pero no tiene en cuenta el diseño de los mismos, pudiendo ocasionar falsos positivos. Se podría emplear un control estadístico del tamaño de los trazos.

## 6. Anexos

### ANEXO 1: Escritura japonesa

El japonés, una lengua hablada principalmente dentro del archipiélago de Japón por más de 100 millones de persona, es una lengua bastante particular. Históricamente, fue una lengua transmitida por el habla. No hubo tipo de escritura oficial alguna hasta la llegada de la cultura china en el siglo III d.C., con lo que la aristocracia japonesa de la época intentó adaptar a lo largo de siglos un gran repertorio de los caracteres chinos (*hanzi*) para la escritura, aunque mezclando la simbología de los caracteres con sus propias palabras para cada símbolo. A lo largo de los siglos, además de ir perfeccionando y adaptando más su propia lengua a los caracteres chinos, fueron creando caracteres

propios como los *kana* (かな) (仮名), simplificaciones de kanji para representar todos sus sonidos silábicos, de aspecto más simple y sencillo que los kanji.

Los *kana* se pueden dividir en dos bloques: *hiragana* (ひらがな) (平仮名) y *katakana* (カタカナ) (片仮名). Los caracteres *hiragana* se usa principalmente para escribir palabras japonesas, partículas gramaticales y desinencias verbales, mientras que los caracteres *katakana* se emplea principalmente para la escritura de palabras extranjeras y onomatopeyas.

Actualmente, en la lengua japonesa se conoce la existencia de más de 10.000 kanji, 1.945 de ellos son reconocidos como obligatorios de aprendizaje durante la enseñanza de los jóvenes japoneses divididos en 7 niveles: 6 niveles que se enseñan en la educación primaria japonesa, compuesto en total por 1.006 kanji, y un nivel extra de 939 kanji enseñados durante la educación secundaria, sin orden alguno preestablecido.

Los kanji contienen múltiples formas de lectura por carácter, englobando sus lecturas en dos grupos:

Lecturas chinas (おんよみ) (音読み) y lecturas japonesas (くんよみ) (訓読み). La principal diferencia entre ambos grupos de lectura radica en que las lecturas chinas se usan en palabras compuestas mientras que las lecturas japonesas se usan en palabras simples, aunque no se cumple para todos los casos.

Los kanji se construyen a partir de trazos. Para hacerse una idea, un trazo correspondería al hecho de coger un lápiz, apoyar la punta sobre la hoja de un papel, mover el lápiz realizando un trazado (normalmente recto) y levantar el lápiz del papel. La escritura de los kanji sigue un orden básico de escritura [40]:

- **Regla 1:** Los kanji tienen que tener el mismo tamaño y tienen que mantener el espacio ocupado por un cuadrado.



- **Regla 2:** Se sigue el siguiente orden básico de trazos:

1. Los kanji se escriben de izquierda a derecha:



2. Los kanji se escriben de arriba a abajo:



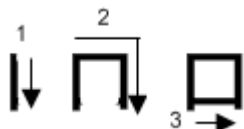
3. Dibuja los trazos desde arriba hacia abajo como se muestra a continuación:



4. Dibuja los trazos desde la izquierda hacia la derecha, como se muestra a continuación:



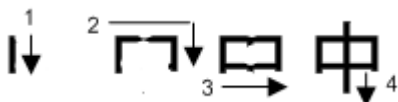
5. Los cuadrados se dibujan como se muestra a continuación:



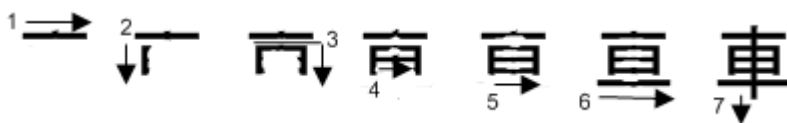
6. Cuando hay una figura dentro del cuadrado, la línea que cierra el cuadrado se dibuja el último, como se muestra a continuación:



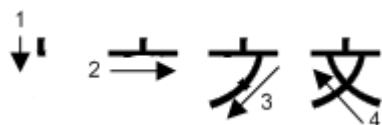
7. Si un trazo cruza a través de un cuadrado, el trazo que lo cruza se dibuja el último, como se muestra a continuación:



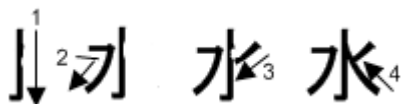
En general, se escriben primero los trazos horizontales si están cortados en su totalidad por un trazo vertical. Ejemplo:



8. Si existen trazos diagonales cruzados, se dibujan primero los inclinados a la izquierda y después los inclinados a la derecha, como se muestra a continuación:



9. Si hay un trazo divisor central, primero se dibuja la parte central, luego sus adyacentes a la izquierda y después los de la derecha, tal y como se muestra a continuación:



## ANEXO 2: Diccionario de datos

Para la realización del proyecto era interesante ofrecer una base de datos nutrida de información sobre los caracteres japoneses y su vocabulario. Para ello se decidió utilizar unos diccionarios de kanji y de vocabulario en XML disponibles en Internet. El diccionario de kanji se le conoce con el nombre de *KANJIDIC2 (KANJI DICTIONARY 2)* [32], mientras que el diccionario de vocabulario se le conoce como *JMdict (Japanese-Multilingual dictionary)* [33]. Estos archivos son proyectos, entre otros, inicializados por Jim Breen en 1991 como un *hobby* que ha ido creciendo hasta convertirse en todo un referente dentro de los diccionarios libres disponibles referentes al

aprendizaje de japonés, teniendo diccionarios para los kanji, el vocabulario, incluyendo vocabulario de nombres de personas, topónimos y otros diccionarios técnicos, radicales de kanji, hasta una recopilación de obras *Ukiyo-e*, obras de arte de tradición japonesa. Estos proyectos se siguen actualizando y ampliando gracias a la ayuda de gente desinteresada.

## KANJIDIC2

El proyecto *KANJIDIC* tiene como meta reunir toda la información comprensible sobre los kanji japoneses. *KANJIDIC2* es la evolución a este proyecto por el mismo autor. Esta versión destaca por estar completamente en *XML*, formato ideal para la lectura de datos sin tener que interpretar los datos de cada línea como sucedía en la versión antigua. *KANJIDIC2* contiene información de los 6.335 kanji establecidos por el estándar *JIS X 0208 (Japanese Industrial Standard X 0208)* [35], los 5.801 kanji establecidos por el estándar *JIS X 0212* [36] y los 3.693 kanji establecidos por el estándar *JIS X 0213* [37], aunque solamente 952 de estos kanji son realmente nuevos en comparación al estándar *JIS X 0212*). Este diccionario contiene la siguiente información interesada para el proyecto:

- El propio kanji en codificación *UTF-8 (8-bit Unicode Transformation Format)*.
- El radical del kanji.
- El grado *JLPT* [38] del kanji. Hay que tener en cuenta que los kanji que tienen grado *JLPT* son una pequeña parte del listado de kanji ofrecido por el diccionario (2.230 kanji de los más de 10.000 ofrecidos). Se han elegido estos por cuestiones prácticas y porque son los necesarios para una buena base de japonés, incluso a nivel experto.
- El número de trazos por kanji.
- Las lecturas chinas y japonesas del kanji.
- Los significados del kanji, principalmente en inglés, aunque los más comunes con traducción en castellano.

Además, contiene otra información interesante para la especialización del estudio de kanji, pero que no se ha podido aprovechar para el proyecto. Toda la información deseada ha sido insertada en nuestra base de datos para una futura utilización más rápida y simple.

## JMdict

El proyecto *JMdict* tiene como meta *la creación de un diccionario libre japonés-inglés de forma legible para las máquinas* [33]. *JMdict* es la evolución directa del proyecto *EDICT*, iniciado el año 1991, transformando toda información mostrada en forma plana por *tags XML*. Actualmente, reúne información de más de 128.000 entradas. Cada entrada contiene la información siguiente:

- La palabra clave con kanji, aunque no por ello única por cada entrada. Si se muestran múltiples palabras, se organizan de mayor a menor, de acuerdo a su usabilidad.
- La lectura en kana de cada entrada de kanji, o la propia palabra clave en sí en kana, si solamente está compuesta por kana. Como con las palabras clave con kanji, las múltiples lecturas por entrada se ordenan de mayor a menor, de acuerdo a su usabilidad.
- Información general de cada entrada como entidad única, con datos como la fecha de creación o la lengua original.
- Elementos de significado, conteniendo las traducciones en inglés equivalentes o glosas de las palabras clave.

Para el proyecto nos ha interesado coger la información de las entradas con palabras claves de mayor uso cotidiano, catalogadas como de alto uso cotidiano por toda una serie de diccionarios en los que se basa la información de las entradas. Estas entradas, unas 29.594 entradas, son almacenadas en base de datos para su uso posterior de forma más rápida y simple.

### **ANEXO 3: Modificaciones internas de Moodle**

Para la creación del foro apto para soportar temas multigrupo e independientes, se tuvieron que realizar cambios en el módulo del foro (*/mod/forum*) a nivel de código. Aquí se muestran los cambios realizados, a partir del uso de la aplicación *diffUtils* para *Windows* [41]. Este programa toma como parámetros el archivo modificado y el archivo original, y devuelve un texto donde se indica, primero, las líneas donde se producen los cambios en ambos archivos y después el bloque de texto modificado. Para más información sobre los comandos, ir a la documentación web de la aplicación:

- **Agregar fichero *teacherdiscussionlib.php* al módulo.**
- **Modificaciones en *lib.php* (*diff -bwB "forumA/lib.php" "forumB/lib.php"*):**

```
3983,3990d3982
< // Modificación para FIC: Para profesores y administradores.
< // Eliminar de la base de datos el permiso para discutir en este tema si existe
< if($discussion->groupid == -1)
< {
<     delete_records("forum_teacher_discussion","discussion",$discussion->id);
< }
< // Fin modificación
<
4483,4486c4475
< // Modificación para FIC: Para estudiantes.
< // Comprueba que un alumno pueda contestar una discusión para todos los
participantes
<     return form_student_can_post($discussion);
< // Fin modificación
---
>     return false;
4751,4754d4739
< // Modificación para FIC: Para todos los participantes.
< // Realizamos un recuento de respuestas dependiendo de si el usuario esta
situado en un grupo concreto o no tiene asignado ningún grupo.
< if (!$currentgroup)
< {
4761,4771c4746
< }
< else
< {
<     if ($numdiscussions > 1000) {
<         // saves some memory on sites with very large forums
<         $replies = forum_count_discussion_replies_in_groups($forum->id,
$currentgroup, $sort, $maxdiscussions, $page, $perpage);
<     } else {
<         $replies = forum_count_discussion_replies_in_groups($forum->id,
$currentgroup);
<     }
< }
< // Fin modificación
---
>
4773,4776d4747
< // Modificación para FIC: Para todos los participantes.
< // Dependiendo de si esta asignado a un grupo o no, se procedera al recu
ento de respuestas de discusiones idoneo.
< if (!$currentgroup)
```

```

<      {
4778,4783c4749
<      }
<      else
<      {
<          $replies = forum_count_discussion_replies_in_groups($forum->id,$
currentgroup);
<      }
<      // Fin modificacion
---
>
4952,4963d4917
<
<      // Modificacion para FIC: Para todos los publicos, principalmente estudian
tes.
<      // Obtiene las respuestas de las discusiones divididas en grupos si el usu
ario esta ubicado en un grupo. Si no, lo muestra todo.
<      $currentgroup = groups_get_activity_group($cm);
<      $groupmode = groups_get_activity_groupmode($cm);
<      $posts = NULL;
<      if($currentgroup && $groupmode == SEPARATEGROUPS)
<      {
<          $posts = forum_get_all_discussion_posts_in_groups($discussion->id, $sort
, $currentgroup, $forumtracked);
<      }
<      if ($posts == NULL)
<      {
4965,4967d4918
<      }
<      // Fin modificacion
<
6885,6888d6835
< // Modificacion para FIC: Para administradores.
< // Libreria donde se almacenaran todas las funciones relacionadas con la tabla
'forum_teacher_discussion'
< require_once('teacherdiscussionlib.php');
< // Fin modificacion

```

- **Modificaciones en post\_form.php (diff -bwB "forumA\post\_form.php" "forumB\post\_form.php"):**

```

44c44,45
<          } else if (isset ( $forum->forcesubscribe ) && $forum->forcesubscribe !=
FORUM_DISALLOWSUBSCRIBE || has_capability ( 'moodle/course:manageactivities',
$coursecontext )) {
---
>          } else if (isset($forum->forcesubscribe)&& $forum->forcesubscribe !=
FORUM_DISALLOWSUBSCRIBE ||
>          has_capability('moodle/course:manageactivities', $coursecontext)) {
93,102d93
<
<      }
<
<      // Modificacion para FIC: Se agrega un campo nuevo en el formulario de
creacion de discusiones para profesores y administradores.
<      // El checkbox sirve para dar permiso a los alumnos a contestar los temas
abiertos a todos los grupos
<      // de los profesores y administradores.
<      if (has_capability ('moodle/legacy:editingteacher', $coursecontext )
<          && $post->parent == 0)
<      {
<          $mform->addElement('checkbox','certificate',
get_string('certificate','forum'));
104d94
<      // Fin modificacion
143c133,134
<      if (($data['timeend'] != 0) && ($data['timestart'] != 0) &&

```



```

$data['timeend'] <= $data['timestart']) {
---
>         if (($data['timeend']!=0) && ($data['timestart']!=0)
>         && $data['timeend'] <= $data['timestart']) {

```

- **Modificaciones en post.php (*diff -bwB "forumA\post.php" "forumB\post.php"*):**

```

158,161d157
<         // Modificacion para FIC: Para alumnos.
<         // Comprobar que el alumno puede contestar en un tema del profesor o
administrador para todos los participantes.
<         if (!form_student_can_post($discussion))
<         {
163,164d158
<         }
<         // Fin Modificacion
504,520d497
<         // Modificacion para FIC: Para administradores y profesores.
<         if (has_capability ( 'moodle/legacy:editingteacher', $coursecontext ))
<         {
<                 $certificate = get_record('forum_teacher_discusion','id',
$discussion->id);
<                 // Si se ha desmarcado la casilla de certificara los
alumnos contestar el tema, eliminar el permiso de la base de datos
<                 if(empty($fromform->certificate) && $certificate)
<                 {
<                         delete_records('forum_teacher_discussion','id',
$discussion->id);
<                 }
<                 // Si se ha marcado la casilla de certificar a los
alumnos contestar el tema, y antes no tenian dicho permiso, agregar el permiso a la base
de datos
<                 elseif (!empty($fromform->certificate) && !$certificate)
<                 {
< forum_add_student_discussion_permission($discussion);
<                 }
<         }
<         // Fin modificacion
631,638d607
<         // Modificacion para FIC: Para profesores
<         // Si se ha marcado la casilla para dar permiso a los alumna
contestar los temas para todos los participantes, actualizar la base de datos.
<         if($fromform->certificate)
<         {
<                 forum_add_student_discussion_permission($discussion);
<         }
<         // Fin modificacion
775,806d743
<         // Modificacion para FIC: Para profesores y administradores.
<         if (has_capability ( 'moodle/legacy:editingteacher', $coursecontext ))
<         {
<                 // Si esta editando la respuesta raiz de un tema sin grupo asignado,
<                 // comprobar si los alumnos tienen autorizacion para escribir y marcarlo en la
casilla si es asi.
<                 if (!empty($edit) && empty($post->parent))
<                 {
<                         // Obtenemos informacion de la discusion
<                         $discussion = get_record('forum_discussions', 'id', $post-
>discussion);
<                         // Si la discusion o tiene grupo asignado, comprobamos si el
alumno tiene autorizacion para agregar respuestas
<                         if($discussion->groupid == -1)
<                         {
<                                 $certificate = form_student_can_post($discussion);

```

```

<
<
<
<
< // HACK_ALERT: Esto no deberia hacerse aqui, pero no hay
otra forma mas limpia de hacerlo.
< // Si el tema a editar es el de un alumno, que
desaparezca el checkbox.
< $mform_post->_form->removeElement('certificate');
<
<
< }
< // Si esta creando un tema nuevo para un grupo en concreto, hacer que
desaparezca el checkbox
< elseif(!empty($forum) && !empty($post->groupid))
< {
< // HACK_ALERT: Esto no deberia hacerse aqui, pero no hay otra
forma mas limpia de hacerlo.
< // Si el tema a crear por un profesor o administrador va a un
grupo en concreto,
< // que desaparezca el checkbox.
< $mform_post->_form->removeElement('certificate');
<
< }
< }
< // Fin modificacion
<
838,845c775
<
< array()+
< // Modificiacion para FIC: Para profesores y
administradores.
< // Agrega la informacion del certificado de alumnos (si
es necesario)
< (isset($certificate)?
< array('certificate'=>$certificate?1:0):
< array())
< // Fin modificacion
< );
---
> array()));

```

- **Modificaciones en version.php (diff -bwB "forumA\version.php" "forumB\version.php"):**

```

8,11c8
< // Modificacion para FIC: Para administradores
< // Actualizar la version a esta fecha para que se ejecute el parche de actualizacion
< $module->version = 2009030200;
< // Fin modificacion
---
> $module->version = 2007101512;

```

- **Modificaciones en db/upgrade.php (diff -bwB "forumA\db\upgrade.php" "forumB\db\upgrade.php"):**

```

86,107d85
< // ModificaciA3n para FIC: Para administradores.
< // Agregar la tabla 'forum_teacher_discussion' a la base de datos para el correcto
funcionamiento del foro.
< if ($result && $oldversion < 2009030200)
< {
< notify('Create table \'forum_teacher_discussion\' for making forum working
correctly...', 'notifysuccess');
<
< // Definimos la tabla a agregar al instalador XML
< $table = new XMLDBTable('forum_teacher_discussion');
< // Creamos los campos de la tabla
< $table->addFieldInfo('id', XMLDB_TYPE_INTEGER, '10', XMLDB_UNSIGNED,

```

```

XMLDB_NOTNULL, XMLDB_SEQUENCE);
<
    $table->addFieldInfo('discussion',XMLDB_TYPE_INTEGER, 10, XMLDB_UNSIGNED,
XMLDB_NOTNULL, null, null, null, null, 'id');
<
    // Asignamos la clave primaria
<
    $table->addKeyInfo('PKid', XMLDB_KEY_PRIMARY, array('id'));
<
    // Asignamos la clave foránea
<
    $table->addKeyInfo('FKdiscussion', XMLDB_KEY_FOREIGN_UNIQUE, array('discussion'),
'forum_discussion','id');
<
    // Asignamos un comentario de la tabla
<
    $table->setComment('Students can reply in discussions w/out assigned group');
<
<
    // Ejecutar la inserción de la tabla
<
    $result = $result && create_table($table);
<
    }
<
    // Fin modificación

```

- **Modificaciones en db/install.xml (diff -bwB "forumA\db\install.php" "forumB\db\install.php"):**

```

2c2
< <XMLDB PATH="mod/forum/db" VERSION="20090302" COMMENT="XMLDB file for Moodle mod/forum"
---
> <XMLDB PATH="mod/forum/db" VERSION="20071010" COMMENT="XMLDB file for Moodle mod/forum"
151c151
<
    <TABLE NAME="forum_track_prefs" COMMENT="Tracks each users untracked forums"
PREVIOUS="forum_read" NEXT="forum_teacher_discussion">
---
>
    <TABLE NAME="forum_track_prefs" COMMENT="Tracks each users untracked forums"
PREVIOUS="forum_read">
164,173d163
<
    <TABLE NAME="forum_teacher_discussion" COMMENT="Students can reply in discussions
w/out assigned group" PREVIOUS="forum_track_prefs">
<
    <FIELDS>
<
    <FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true" UNSIGNED="true"
SEQUENCE="true" ENUM="false" NEXT="discussion"/>
<
    <FIELD NAME="discussion" TYPE="int" LENGTH="10" NOTNULL="true" UNSIGNED="true"
SEQUENCE="false" ENUM="false" PREVIOUS="id"/>
<
    </FIELDS>
<
    <KEYS>
<
    <KEY NAME="pkid" TYPE="primary" FIELDS="id" NEXT="fkdiscussion"/>
<
    <KEY NAME="fkdiscussion" TYPE="foreign-unique" FIELDS="discussion"
REFTABLE="forum_discussion" REFFIELDS="id" PREVIOUS="pkid"/>
<
    </KEYS>
<
    </TABLE>

```

## 7. Bibliografía y referencias

A continuación se muestra toda una serie de referencias a libros y sitios web utilizados para el desarrollo del proyecto. Primero los comentados a lo largo de la memoria.

- [1] J. Cabero. 2006. Bases pedagógicas del *e-learning*. Disponible en URL: <http://www.uoc.edu/rusc/3/1/dt/esp/cabero.html>
- [2] Página oficial de Blackboard, disponible en URL (inglés): <http://www.blackboard.com/>
- [3] Wikipedia. WebCT, disponible en URL (inglés): <http://en.wikipedia.org/wiki/WebCT>
- [4] Página oficial de Desire2Learn, disponible en URL (inglés): <http://www.desire2learn.com/>
- [5] Página oficial de Dokeos, disponible en URL (inglés): <http://www.dokeos.com/>
- [6] Página oficial de ATutor, disponible en URL (inglés): <http://www.atutor.ca/>
- [7] Página oficial de Sakai Project, disponible en URL (inglés): <http://sakaiproject.org/portal>
- [8] Página oficial de Moodle, disponible en URL: <http://moodle.org/>
- [9] Página oficial de Apache, disponible en URL (inglés): <http://www.apache.org/>
- [10] Página oficial de PHP, disponible en URL (inglés): <http://www.php.net/>
- [11] Página oficial de ADOdb, disponible en URL (inglés): <http://adodb.sourceforge.net/>
- [12] Página oficial de MySQL, disponible en URL (inglés): <http://www.mysql.com/>
- [13] Página oficial de XAMPP, disponible en URL (inglés): <http://www.apachefriends.org/en/xampp.html>
- [14] MoodleDocs. 2009. User site capacities. Disponible en URL: [http://docs.moodle.org/en/User\\_site\\_capacities](http://docs.moodle.org/en/User_site_capacities)
- [15] Página oficial de Mozilla Firefox, disponible en URL: <http://www.mozilla-europe.org/es/firefox/>
- [16] Página oficial de Internet Explorer, disponible en URL: <http://www.microsoft.com/spain/windows/internet-explorer/default.aspx>
- [17] Página oficial de Safari, disponible en URL: <http://www.apple.com/safari/>
- [18] Página oficial de Google Chrome, disponible en URL: <http://www.google.com/chrome>
- [19] Página oficial de Aptana Studio, disponible en URL (inglés): <http://aptana.com/studio>
- [20] Página oficial de jQuery, disponible en URL (inglés): <http://jquery.com/>
- [21] Página oficial de jCarousel, disponible en URL (inglés): <http://sorgalla.com/jcarousel/>
- [22] Página oficial de jTemplates, disponible en URL (inglés): <http://jtemplates.tpython.com/>
- [23] Página oficial de UIBlock, disponible en URL (inglés): <http://malsup.com/jquery/block/>
- [24] Página oficial de Adobe Flash, disponible en URL: <http://www.adobe.com/products/flash/>
- [25] Listado de plug-ins oficiales de Moodle, disponible en URL (inglés): <http://moodle.org/mod/data/view.php?id=6009>
- [26] Información de plug-in *COVCELL Audio/Video Conferencing Tool*, disponible en URL (inglés): <http://moodle.org/mod/data/view.php?id=13&rid=1190&filter=1>
- [27] Wikipedia. Programación por capas, disponible en URL:

[http://es.wikipedia.org/wiki/Programaci%C3%B3n\\_por\\_capas](http://es.wikipedia.org/wiki/Programaci%C3%B3n_por_capas)

[28] Frank Ralf. Checkboxes: Show only selected, disponible en URL (inglés):

[http://docs.moodle.org/en/User:Frank\\_Ralf/JavaScript1](http://docs.moodle.org/en/User:Frank_Ralf/JavaScript1)

[29] J.Tokuno, N. Inami, S. Matsuda, M. Nakai, H. Shimodaira, S. Sagayama. 2002. Context-dependent Substroke Model for HMM-based On-line Handwriting Recognition. Disponible en URL (inglés): <http://www2.computer.org/portal/web/csdl/doi/10.1109/IWFHR.2002.1030888>

[30] Wikipedia. Modelo Oculto de Márkov, disponible en URL:

[http://es.wikipedia.org/wiki/Modelo\\_oculto\\_de\\_M%C3%A1rkov](http://es.wikipedia.org/wiki/Modelo_oculto_de_M%C3%A1rkov)

[31] Wikipedia. Autómata finito, disponible en URL: [http://es.wikipedia.org/wiki/Aut%C3%B3mata\\_finito](http://es.wikipedia.org/wiki/Aut%C3%B3mata_finito)

[32] Página oficial de KANJIDIC2, disponible en URL (inglés):

<http://www.csse.monash.edu.au/~jwb/kanjdic2/>

[33] Página oficial de JMDict, disponible en URL (inglés):

[http://www.csse.monash.edu.au/~jwb/edict\\_doc.html](http://www.csse.monash.edu.au/~jwb/edict_doc.html)

[34] Wikipedia. JIS X 0208, disponible en URL (inglés): [http://en.wikipedia.org/wiki/JIS\\_X\\_0208](http://en.wikipedia.org/wiki/JIS_X_0208)

[35] Wikipedia. JIS X 0212, disponible en URL (inglés): [http://en.wikipedia.org/wiki/JIS\\_X\\_0212](http://en.wikipedia.org/wiki/JIS_X_0212)

[36] Wikipedia. JIS X 0213, disponible en URL (inglés): [http://en.wikipedia.org/wiki/JIS\\_X\\_0213](http://en.wikipedia.org/wiki/JIS_X_0213)

[37] Wikipedia. Nihongo nôryoku shiken, disponible en URL:

[http://es.wikipedia.org/wiki/Nihongo\\_n%C5%8Dryoku\\_shiken](http://es.wikipedia.org/wiki/Nihongo_n%C5%8Dryoku_shiken)

[38] Wikipedia. Marco Común Europeo de Referencia para las lenguas, disponible en URL:

[http://es.wikipedia.org/wiki/Marco\\_Com%C3%B3n\\_Europeo\\_de\\_Referencia\\_para\\_las\\_lenguas](http://es.wikipedia.org/wiki/Marco_Com%C3%B3n_Europeo_de_Referencia_para_las_lenguas)

[39] Y. Suzuki, H. Terada, M. Fukuda. 2007. Teaching and Learning Writing Skills in Japanese Using the Web Remote Communication Activity (Spain and Japan) on an Internet Discussion Board. *Symposium on Japanese Language Education*.

[40] J. Y. Mitamura, Y. K. Mitamura. 1997. Let's Learn Kanji: An introduction to radicals, components, and 250 very basic kanji. *Kodansha Europe Ltd*. Pág. 5-10

[41] Página oficial de diffUtils para Windows, disponible en URL:

<http://gnuwin32.sourceforge.net/packages/diffutils.htm>

[42] MoodleDocs. 2009. PHP Settings by Moodle version, disponible en URL (inglés):

[http://docs.moodle.org/en/PHP\\_settings\\_by\\_Moodle\\_version](http://docs.moodle.org/en/PHP_settings_by_Moodle_version)

[43] Página de descarga de Adobe Flash Player, disponible en URL:

<http://www.adobe.com/es/products/flashplayer/>

[44] MoodleDocs. 2009. Development:Javascript guidelines, disponible en URL:

[http://docs.moodle.org/en/Development:JavaScript\\_guidelines](http://docs.moodle.org/en/Development:JavaScript_guidelines)

Otra documentación útil:

2004. Technical Evaluation of selected Learning Management Systems. The Open Polytechnic of New Zealand. Disponible en URL (inglés): <https://eduforge.org/docman/view.php/7/18/lms%2520technical%2520evaluation%2520-%2520may04.pdf/>

V. G. Verdú. 2002. Utilidad de las nuevas tecnologías en lengua alemana para universitarios. Experiencias didácticas desde un enfoque constructivista. Disponible en URL:

<http://dialnet.unirioja.es/servlet/articulo?codigo=2881606>

M. D. Catrillo de Larreta-Azelain, J. C. G. Cabrero, G. Ruipérez. 2003. E-learning en la enseñanza del alemán como lengua extranjera. Disponible en URL: <http://dialnet.unirioja.es/servlet/articulo?codigo=2881611>

M. Nakai, N. Akira, H. Shimodaira, S. Sagayama. 2001. Substroke Approach to HMM-based On-line Kanji Handwriting Recognition. Disponible en URL (inglés): <http://www.cstr.ed.ac.uk/downloads/publications/2001/Nakai2001ICDAR.pdf>

C. C. Tappert, C. Y. Suen, T. Wakahara. 1990. The State of the Art in On-line Handwriting Recognition. Disponible en URL (inglés): <http://csis.pace.edu/~ctappert/dps/pdf/pen-tappert.pdf>

C. Darie, E. Balanescu. 2008. Beginning PHP and MySQL E-Commerce From Novice to Professional 2<sup>nd</sup> Edition. *Apress*.

**Resumen.** El trabajo descrito en esta memoria es el resultado de un proyecto de colaboración universitaria entre la Universidad Autónoma de Bellaterra, la Universidad Complutense de Madrid y la Keio University Shonan Fujisawa Campus de Japón, cuya finalidad es la de ayudar a estudiantes de lenguas japonesa y española en horas extra escolares a mejorar y practicar sus conocimientos elementales de las lenguas de estudio mediante el empleo de tecnologías Web. Se describe aquí la plataforma seleccionada para el desarrollo de un foro conversacional y otras utilidades que ayudará y reforzará a los estudiantes los conocimientos del idioma japonés suministrados en horarios de clase.

**Resum.** El treball descrit en aquesta memòria es el resultat d'un projecte de col·laboració universitària entre la Universitat Autònoma de Bellaterra, la Universitat Complutense de Madrid i la Keio University Shonan Fujisawa Campus de Japó, amb la finalitat d'ajudar a estudiants de llengües japonesa i espanyola en horari extraescolar a millorar i practicar el seus coneixements elementals de les llengües de estudi mitjançant la utilització de tecnologies Web. Es descriu aquí la plataforma seleccionada per al desenvolupament d'un fòrum conversacional i d'altres utilitats que ajudaran i reforçaran als estudiants els coneixements de l'idioma japonès subministrats en horari de classe.

**Abstract.** The described work in this report is the outcome from university collaboration between Autonomous University of Bellaterra, Universidad Complutense of Madrid and Keio University Shonan Fujisawa Campus from Japan, in the pursuit of help spanish and japanese language students in out-of-school timetable to improve and to practice their knowledge with Web technologies collaboration.

Here is described the selected platform for a conversational forum development and another utilities that will help and will strengthen japanese language students' knowledge, gained in class time.